NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ANALYSIS OF MULTIRATE RANDOM SIGNALS

by

Dimitrios Koupatsiaris

December 2000

Supervisor:	Charles W. Therrien
Committee Member:	Roberto Cristi
Committee Member:	Xiaoping Yun

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, Va 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE December 2000	3. REPORT TYPE AND DATES COVERED Engineer's Thesis

4. TITLE AND SUBTITLE Analysis of Multirate Random Signals	5. FUNDING NUMBERS
6. AUTHOR(S) Koupatsiaris, Dimitrios	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT(*maximum 200 words*)

Multirate digital signal processing techniques have been developed in the recent years for a wide range of applications, such as speech and image compression, digital audio, statistical and adaptive signal processing, numerical solution of differential equations and many other fields.

The purpose of this thesis is to extend optimal filtering techniques to random signals sampled at different rates. In particular, two major problems are considered: (1) optimal filtering of two sets of observations at different sampling rates as a multirate Wiener filter, and (2) linear prediction on successive samples of a random process. In the first problem it is shown that the standard Wiener filter can be extended to the multirate case, while preserving its optimality. In the second problem it is shown that multichannel linear prediction on successive samples of a process, yields orthogonal uncorrelated innovations.

14. SUBJECT TERMS Multirate signal analysis, Estimation, Wiener filter	15. NUMBER OF PAGES 93
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

THIS PAGE INTENTIONALLY LEFT BLANK

ANALYSIS OF MULTIRATE RANDOM SIGNALS

Dimitrios A. Koupatsiaris
Lieutenant, Hellenic Navy
B.S., Hellenic Naval Academy, 1992

Submitted in partial fulfillment of the
requirements for the degree of

ELECTRICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
December 2000

Author: Dimitrios A. Koupatsiaris

Approved by: Charles W. Therrien
Charles W. Therrien, Supervisor

Roberto Cristi
Roberto Cristi, Committee Member

Xiaoping Yun, Committee Member

Jeffrey B. Knorr, Chairman
Department of Electrical & Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Multirate digital signal processing techniques have been developed in the recent years for a wide range of applications, such as speech and image compression, digital audio, statistical and adaptive signal processing, numerical solution of differential equations and many other fields.

The purpose of this thesis is to extend optimal filtering techniques to random signals sampled at different rates. In particular, two major problems are considered: (1) optimal filtering of two sets of observations at different sampling rates as a multirate Wiener filter, and (2) linear prediction on successive samples of a random process. In the first problem it is shown that the standard Wiener filter can be extended to the multirate case, while preserving its optimality. In the second problem it is shown that multichannel linear prediction on successive samples of a process, yields orthogonal uncorrelated innovations.

THIS PAGE INTENTIONALLY LEFT BLANK

DISCLAIMER

The computer programs in the Appendix are supplied on an “as is” basis, with no warrantees of any kind. The author bears no responsibility for any consequences of using these programs.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to express my gratitude and appreciation for the assistance provided in the development of this thesis to Dr. Charles W. Therrien and Dr. Roberto Cristi. The completion of this project relied on their guidance and professional counsel.

I also want to dedicate this work to my wife Penelope for her understanding and support.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In many practical applications of digital signal processing we have to process signals sampled at different rates. For example, in telecommunication systems that transmit and receive various types of signals (e.g. teletype, facsimile, speech, video, etc.), the different sampling rates are related to the bandwidths of the corresponding signals. The process of converting a signal from a given sampling rate to a different one is called *sampling rate conversion.* In turn, systems that process signals at different rates are called *multirate digital signal processing systems.*

The most practical method of sampling rate conversion of a digital signal is to perform it entirely in the digital domain, as a combination of upsampling and downsampling by an integer factor [Ref. 3]. In particular, the process of reducing the sampling rate by an integer factor D (downsampling by D) is called *decimation*, while the process of increasing the sampling rate by an integer factor U is called *upsampling* (it is also called *expansion* or *interpolation*).

A. PREVIOUS RELATED RESEARCH

During the past decade a very important area of research has been in *Multiresolution* and *Multirate Digital Signal Processing.* Two very successful approaches to this problem have been developed: one based on wavelet decomposition [Ref. 1], and another based on the multiresolution Kalman filtering [Ref. 4] and [Ref. 5]. In the following sections, we provide an outline of both methods and their applications in multirate signal processing.

1. Wavelet Decomposition

A wavelet is a "small wave," a wave that has its energy concentrated in time to give a tool for the analysis of transient, non stationary, or time-varying phenomena [Ref. 1]. The property stated above, allows the wavelet to perform analysis in both frequency and time with a flexible mathematical foundation. For that purpose, the wavelet decomposition uses two functions; the *scaling function* $\varphi(t)$ that is responsible to track the signal in time, and the *mother wavelet* $\psi(t)$, related to $\varphi(t)$, that is the function that performs analysis of a signal using multiple resolutions. Based on the two functions defined above, a large number of signals can be decomposed as an expansion also called the Discrete Wavelet Transform (DWT):

$$f(t) = \sum_{k=-\infty}^{\infty} c_k \cdot \varphi(t-k) + \sum_{k=-\infty}^{\infty} \sum_{j=0}^{\infty} d_{j,k} \cdot \psi(2^j t - k). \tag{I.1}$$

The functionality of Equation I.1 can be described with the use of vector spaces. We call $\mathcal{S}$ the vector space of signals where any $f(t) \in \mathcal{S}$ can be expressed as

$$f(t) = \sum_k a_k \cdot \varphi_k(t). \tag{I.2}$$

Then, the set of functions $\varphi_k(t)$ is called an expansion set for $\mathcal{S}$. If the representation is unique, the set is a basis. If one starts with the basis set and defines the space $\mathcal{S}$ as the set of all functions $f(t)$ that can be expressed by Equation I.2, this set is called the span of the basis set. Then, we define a set of scaling functions in terms of integer translation of the basic scaling function by

$$\varphi_k(t) = \varphi(t-k) \tag{I.3}$$

where k is an integer ($k \in \mathbf{Z}$). One can generally increase the size of the subspace spanned by changing the time scale of the basis functions. However, the important features of a signal can be described or parameterized more efficiently by defining a set of functions

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \tag{I.4}$$

that span the "difference" subspaces or orthogonal complement spaces, defined as the subspaces between the spaces spanned by the various scales of the scaling function.

The selection of the mother wavelet is not unique, however $\psi(t)$ is usually chosen such that the set of functions in all resolutions comprises an *orthonormal set*, where for a complex basis set:

$$\langle \psi_k(t), \psi_l(t) \rangle = \int \psi_k(t) \cdot \psi_l^*(t) dt = 0 \tag{I.5}$$

and

$$\langle \psi_k(t), \psi_k^*(t) \rangle = 1. \tag{I.6}$$

The need for wavelets came from the inherent weakness of the Fourier-based methods to analyze signals with edges, time-varying signals, or broadband signals such as transients. On the other hand, the DWT uses a 2-D expansion set (the indexes j for frequency and k for time) that provides a very close to optimal analysis of a signal, localized in both time and frequency.

A particularly interesting class of problems of multirate signal processing is *multiresolution analysis*, where a signal is decomposed into a lower and an upper frequency component in a recursive fashion [Ref. 6: pp.254-259]. The filter banks that implement this kind of structure are called *tree structured filter banks* and they are very useful in the implementation of wavelet analysis. In particular, one can express $\psi(t)$ and $\varphi(t)$ in terms of filtering functions as follows:

$$\varphi(t) = \sum_n h_0(n) \cdot \sqrt{2} \cdot \varphi(2t - n) \tag{I.7}$$

where n is an integer ($n \in \mathbf{Z}$), and the coefficients $h_0(n)$ are a sequence of real or complex numbers called the scaling function coefficients (or the scaling filter) and $\sqrt{2}$ maintains the norm of the scaling function. Equation I.7 is called the multiresolution analysis (MRA) equation, or the dilation equation. Also

$$\psi(t) = \sum_n h_1(n) \cdot \sqrt{2} \cdot \varphi(2t - n) \tag{I.8}$$

for some set of coefficients $h_1(n)$. From the requirement that $\psi_{j,k}(t)$ span the "difference" subspaces, and the orthogonality of integer translates of the wavelet, the coefficients $h_1(n)$ are required to be related to the scaling function coefficients by

$$h_1(n) = (-1)^n \cdot h_0(1 - n). \tag{I.9}$$

We can then perform an analysis from fine scale to coarse scale (fine to coarse resolution) and obtain the coefficients c_k and $d_{j,k}$ that appear in Equation I.1 by iterating the *tree-structured* sections of Figure 1.

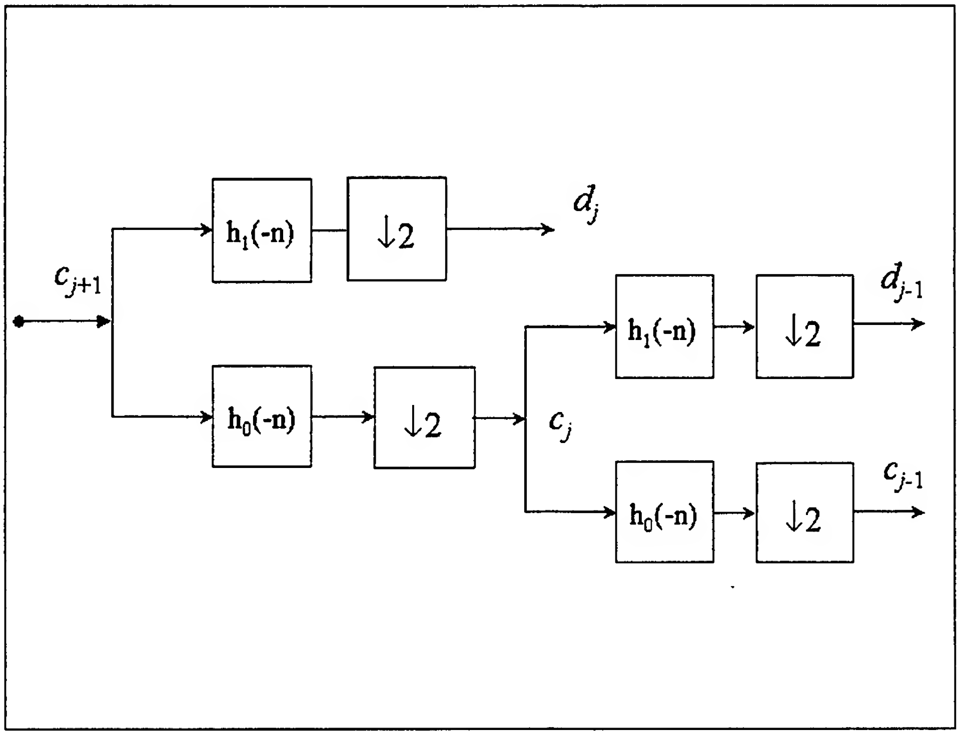

Figure 1. Two-Stage Two-Band Analysis Tree (from [Ref. 1])

Despite the advantages that were previously stated, the wavelet theory can only perform sub-optimal analysis of a signal, whether it is stationary or not. The reason for this is that it does not take into account the signal properties or, in the case of a random signal, the signal statistics. In addition, wavelet theory is specialized in the analysis and synthesis of individual signals, whether they are 1-D or 2-D. However, there is a new class of applications that deals with the synthesis of different signals that descent from the same origin and collected individually (i.e. different sensors), called *information fusion*. This is the kind of problems that the following application attempts to solve.

2. Multiresolution Kalman Filter

Multiresolution Multirate techniques have been developed during the past ten years, based on the Kalman Filtering approach. The result is a recursive algorithm both in time as well as in resolution, which optimally combines estimates and observations at different levels of resolution and sampling rate. The general theory has been presented initially in [Ref. 7], and then refined in [Ref. 4] and [Ref. 5]. The framework presented in this line of research is based on a general solution of a Riccati equation, recursive in time and in resolution.

A different approach has been taken by [Ref. 8], where multiresolution modeling and filtering of time series have been addressed based on a recursive Kalman Filtering approach. It is shown that a time series can be decomposed into a number of models operating at different sampling rates, having independent innovations. Prediction and optimal filtering based on a number of observations at different sampling rates have been presented.

The general Kalman Filtering approach yields a class of estimators which are Infinite Impulse Response (IIR), and attempt to determine the optimal prediction of the time series at time $n+1$, based on all the observations from the initial time (say $n = 0$) to the current observation time n. We say that this approach has an infinite memory, and therefore it necessitates IIR filters to be implemented. However the tendency in applied Digital Signal Processing is to use Finite Impulse Response (FIR) filters, and the approach of this thesis will be to develop a class of optimal predictors based on a finite window of data.

B. THESIS APPROACH

While the previously mentioned approaches have been successful, they are somewhat difficult to interpret in terms of conventional sampling and filtering operations that are the mainstream of signal processing for deterministic signals. The arguments presented in this thesis are based on a number of methods representing decimation and linear filtering. It turns out that a very effective way of representing them is by using linear algebra techniques. In this way each operator is associated with a matrix operation, and the analysis is based on a number of concepts available in the linear algebra literature.

In the first part of this research we formulate the tools for developing the optimal filters for problems with multiple observations at different sampling rates. In particular, we show that a Wide Sense Stationary (WSS) stochastic process can be estimated based on observations at different sampling rates, all affected by measurement noise. The outcome is an optimal linear time-varying FIR filter with periodically changing coefficients. The filter is designed on the basis of a set of equations similar to the Wiener-Hopf equations in the standard Wiener filter setting, derived directly from the orthogonality principle. The resulting formulation is an information fusion application, developed for the case of two observation sequences at different rates, but extendable to more than two sets of observations.

In the second part of the research we use the methods of linear prediction and its multichannel extension in order to perform linear prediction of a random process in multiple resolutions. In particular, we form two separate channels from the even and odd samples of the process, and perform linear prediction on the multichannel process consisting of these two sets of samples. The resulting formulation is a multiple resolution application, extending to capabilities similar to those of wavelet decomposition.

C. THESIS OUTLINE

This thesis report consists of five chapters including the introduction. Chapter II provides an introduction to notation and algebraic manipulations that are used extensively in the work. Chapter III develops the multirate Wiener filter and performs performance comparison with the single rate Wiener filter. Chapter III also provides variations and generalizations of the multirate Wiener filter and performs performance analysis simulations between the two filters. Chapter IV deals with the multichannel and multirate linear prediction. Finally, Chapter V summarizes the work and suggests future directions of research.

THIS PAGE INTENTIONALLY LEFT BLANK

II. MATHEMATICAL PRELIMINARIES

Wiener filtering [Ref. 2: pp.347-354], named after Norbert Wiener (1894-1964), is a general type of optimal linear filtering. It involves the estimation of a *wide-sense stationary* (WSS) random process $s[n]$, called the "desired" process, that cannot be observed directly, from the observation of a jointly stationary random process $x[n]$. The desired random process may represent a signal which is subject to various forms of distortion and interference. In the simplest form of the problem, the goal of the signal processing is to estimate the sequence $s[n]$ with a sequence $\hat{s}[n]$ from the observed sequence $x[n]$ via a linear *finite impulse response* (FIR) filter. Specifically, it is required to use a linear combination of the present value $x[n]$ and the last $P-1$ values to produce an estimate that minimizes the *mean-square error* (MSE). Wiener filtering can also be performed using an infinite impulse response (IIR) filter, however this case will not be considered here.

In order to minimize the MSE, it is necessary and sufficient to satisfy the *orthogonality principle.* Specifically, this principle states that if we define $\varepsilon[n] = s[n] - \hat{s}[n]$ the error in the estimation, the resulting error should be orthogonal to the observations, or

$$E\left\{x[n-i] \cdot \varepsilon^*[n]\right\} = \mathit{0}, \quad \mathit{where} \quad i = 0, 1, \ldots, P-1. \tag{II.1}$$

The filter impulse response that satisfies the orthogonality principle is denoted by the vector $\mathbf{h}$ and is obtained as the solution of the matrix equation

$$\mathbf{R}_{xx} \cdot \mathbf{h} = \tilde{\mathbf{r}}_{sx} \tag{II.2}$$

where $\mathbf{R}_{xx}$ is a Toeplitz matrix formed from the *autocorrelation function* (ACF) of the sequence $x[n]$ and $\mathbf{r}_{sx}$ is a vector formed from the cross-correlation between $s[n]$ and $x[n]$. Equation II.2 is known as the *Wiener-Hopf* equation. The resulting estimate can be written as

$$\hat{s}[n] = \tilde{\mathbf{x}}^{*T} \cdot \mathbf{h} \tag{II.3}$$

while the minimum MSE can be expressed as

$$\sigma_\varepsilon^2 = R_{ss}[0] - \tilde{\mathbf{r}}_{sx}^T \cdot \mathbf{h}^* = R_{ss}[0] - \mathbf{h}^{*T} \cdot \tilde{\mathbf{r}}_{sx} \tag{II.4}$$

where the˜(tilde) operator denotes the reversed of the vector (see Section A below).

The first problem addressed in this thesis, extends the FIR Wiener filter to the case of multirate observation sequences. However, the guiding principles are the same as those discussed above.

A. BASIC DEFINITIONS

1. Observation Vectors

In the analysis of random signals it is convenient to represent finite length sequences as vectors. We can represent a signal $s[k]$ on the interval $n-N+1 \leq k \leq n$ with the vector

$$\mathbf{s}_n = \left[\begin{array}{cccc} s[n-N+1] & s[n-N+2] & \ldots & s[n] \end{array} \right]^T \tag{II.5}$$

and identify it as a *random observation vector* for $s[k]$. Since the values of the signal are random variables and are possibly complex-valued, the vector elements may also be complex-valued random variables.

It is evident from the definition that the length of an observation vector may vary according to our will. In addition, when we have multiple observations, not necessarily of the same length, say from $x[n]$ and $y[n]$, we can define an augmented observation vector as a combination of $\mathbf{x}_n$ and $\mathbf{y}_n$, as follows:

$$\begin{aligned} \mathbf{x}_n &= \left[\begin{array}{cccc} x[n-N+1] & x[n-N+2] & \ldots & x[n] \end{array} \right]^T \\ \mathbf{y}_m &= \left[\begin{array}{cccc} y[m-M+1] & y[m-M+2] & \ldots & y[m] \end{array} \right]^T \\ \mathbf{z} &= \left[\begin{array}{c} \mathbf{x}_n \\ \mathbf{y}_m \end{array} \right]. \end{aligned} \tag{II.6}$$

2. Reversal Operation

In the analysis that follows, it is often necessary to reverse the order of the points in a given vector by turning the vector *upside-down* (see [Ref. 2]). Given a vector $\mathbf{x}_n$ we define its time reversal $\tilde{\mathbf{x}}_n$ as:

$$\mathbf{x}_n = \begin{bmatrix} x[n-N+1] \\ x[n-N+2] \\ \vdots \\ x[n] \end{bmatrix}, \quad \textit{then} \quad \tilde{\mathbf{x}}_n = \begin{bmatrix} x[n] \\ \vdots \\ x[n-N+2] \\ x[n-N+1] \end{bmatrix}. \qquad \text{(II.7)}$$

Likewise, the reversal of a matrix is defined as a matrix with the elements reversed about both its vertical and its horizontal axis. Table I shows some elementary properties of the reversal.

-	Quantity	Reversal
Matrix product	$\mathbf{A}\cdot\mathbf{B}$	$\tilde{\mathbf{A}}\cdot\tilde{\mathbf{B}}$
Matrix inverse	$\mathbf{A}^{-1}$	$(\tilde{\mathbf{A}})^{-1}$
Matrix conjugate	$\mathbf{A}^*$	$\tilde{\mathbf{A}}^*$
Matrix transpose	$\mathbf{A}^T$	$\tilde{\mathbf{A}}^T$

Table I. Properties of Reversal (from [Ref. 2])

3. Correlation Matrices

In this thesis we focus our attention on random processes. In the case that the processes are Gaussian in nature, the first and second moments are sufficient to fully characterize the process. In general, even when the processes under consideration are not Gaussian, the second order moments are still at the basis of the design of optimal linear (Wiener) filters. In particular, we define the *mean* of a random process as: $m_x[n] = E\{x[n]\}$, and the *autocorrelation function* (ACF) of a random process as:

$$R_{xx}[n_1, n_0] = E\{x[n_1] \cdot x^*[n_0]\}. \qquad \text{(II.8)}$$

In the particular case where the process is *wide-sense stationary* (WSS), the mean is a constant m_x and the ACF is a function of the time lag between the samples:

$$R_{xx}[l] = E\left\{x[n] \cdot x^*[n-l]\right\}. \tag{II.9}$$

Similarly we define the *covariance function* as: $C_{xx}[l] = R_{xx}[l] - |m_x|^2$. When a random process has zero mean, the ACF and the covariance function are the same.

Given an observation vector $\mathbf{s}_n$, we define the *correlation matrix* of a zero-mean, (WSS) signal $s[n]$ as

$$\begin{aligned}\mathbf{R}_{ss} &= E\left\{\mathbf{s}_n \cdot \mathbf{s}_n^{*T}\right\} \\ &= E\left\{\begin{bmatrix} s[n-N+1] \\ s[n-N+2] \\ \vdots \\ s[n] \end{bmatrix} \cdot \begin{bmatrix} s^*[n-N+1] & s^*[n-N+2] & \cdots & s^*[n] \end{bmatrix}\right\}.\end{aligned} \tag{II.10}$$

Due to the WSS assumption and the definition of the ACF, we can write $\mathbf{R}_{ss}$ as

$$\mathbf{R}_{ss} = \begin{bmatrix} R_{ss}[0] & R_{ss}[-1] & \cdots & R_{ss}[-N+1] \\ R_{ss}[1] & R_{ss}[0] & \cdots & R_{ss}[-N+2] \\ \vdots & \vdots & \ddots & \vdots \\ R_{ss}[N-1] & R_{ss}[N-2] & \cdots & R_{ss}[0] \end{bmatrix}. \tag{II.11}$$

Observe that $\mathbf{R}_{ss}$ is Toeplitz, since all elements on each principal diagonal are equal. The correlation matrix is also Hermitian symmetric and positive semi-definite, but this is true regardless of the WSS assumption. We can form the correlation matrix from the reversed observation vector as:

$$\tilde{\mathbf{R}}_{ss} = E\left\{\tilde{\mathbf{s}}_n \cdot \tilde{\mathbf{s}}_n^{*T}\right\} = \begin{bmatrix} R_{ss}[0] & R_{ss}[1] & \cdots & R_{ss}[N-1] \\ R_{ss}[-1] & R_{ss}[0] & \cdots & R_{ss}[N-2] \\ \vdots & \vdots & \ddots & \vdots \\ R_{ss}[-N+1] & R_{ss}[-N+2] & \cdots & R_{ss}[0] \end{bmatrix}. \qquad \text{(II.12)}$$

Notice that because of the Hermitian symmetry property of the autocorrelation function ($R_{ss}[-l] = R_{ss}^*[l]$), it follows that

$$\tilde{\mathbf{R}}_{ss} = \mathbf{R}_{ss}^*. \qquad \text{(II.13)}$$

4. Cross-correlation Matrices

For the purpose of representing the joint statistics of two random processes $x[n]$ and $y[n]$, we define the cross-correlation function as:

$$R_{xy}[n_1, n_0] = E\left\{x[n_1] \cdot y^*[n_0]\right\}. \qquad \text{(II.14)}$$

In the case of joint stationarity, the cross-correlation is also a function of the time lag l and the definition becomes

$$R_{xy}[l] = E\left\{x[n] \cdot y^*[n-l]\right\}. \qquad \text{(II.15)}$$

The cross-correlation vector between the signals $s[n]$ and vector of observations $\mathbf{x}_n$ is defined as:

$$\mathbf{r}_{sx} = E\left\{s[n] \cdot \mathbf{x}_n^*\right\} = \begin{bmatrix} R_{sx}[N-1] \\ \vdots \\ R_{sx}[0] \end{bmatrix} \qquad \text{(II.16)}$$

or in its time reversed form as:

$$\tilde{\mathbf{r}}_{sx} = E\left\{s[n] \cdot \tilde{\mathbf{x}}_n^*\right\} = \begin{bmatrix} R_{sx}[0] \\ \vdots \\ R_{sx}[N-1] \end{bmatrix}. \qquad \text{(II.17)}$$

More generally, given two sets of observations represented by the vectors $\mathbf{x}_n$ of length N and $\mathbf{y}_m$ of length M, we define their cross-correlation matrix as:

$$\mathbf{R}_{xy} = E\left\{\mathbf{x}_n \cdot \mathbf{y}_m^{*T}\right\} = \begin{bmatrix} R_{xy}[0] & R_{xy}[-1] & \cdots & R_{xy}[-M+1] \\ R_{xy}[1] & R_{xy}[0] & \cdots & R_{xy}[-M+2] \\ \vdots & \vdots & \ddots & \vdots \\ R_{xy}[N-1] & R_{xy}[N-2] & \cdots & R_{xy}[0] \end{bmatrix}. \tag{II.18}$$

The resulting matrix is not square (in general) and has no particular properties except for the repetition of elements along diagonals.

B. DECIMATION AND UPSAMPLING

The processes of decimation and upsampling are the basic tools of multirate digital signal processing. In the following sections we present two operators that extend sampling rate conversion as algebraic operations.

1. Decimation Matrices

The operations of decimation and upsampling can be represented by appropriate matrix operations. Let

$$\mathbf{s}_n = \begin{bmatrix} s[n-2N+1] & s[n-2N+2] & \ldots & s[n] \end{bmatrix}^T \tag{II.19}$$

represent a sequence $s[n]$, of length $2N$ ending with the observation $s[n]$. In that case,

$$\begin{aligned} \mathbf{s}_e &= \begin{bmatrix} s[n-2N+2] & \ldots & s[n+2] & s[n] \end{bmatrix}^T \\ \mathbf{s}_o &= \begin{bmatrix} s[n-2N+1] & \ldots & s[n+3] & s[n+1] \end{bmatrix}^T \end{aligned} \tag{II.20}$$

represent the even and odd sample observations as shown in Figure 2. The *Decimation Matrix* $\mathbf{D}^{(0)}$ and *Decimation Matrix with Time Delay* $\mathbf{D}^{(1)}$ are then defined as:

$$\mathbf{D}^{(0)} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{II.21}$$

and

$$\mathbf{D}^{(1)} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{II.22}$$

and they are such that

$$\mathbf{s}_e = \mathbf{D}^{(0)} \cdot \mathbf{s}_n \quad \mathit{and} \quad \mathbf{s}_o = \mathbf{D}^{(1)} \cdot \mathbf{s}_n. \tag{II.23}$$

Both matrices $\mathbf{D}^{(0)}$ and $\mathbf{D}^{(1)}$ are of size $N \times 2N$.

2. Upsampling Matrices

Let

$$\mathbf{s}_n = \begin{bmatrix} s[n-N+1] & s[n-N+2] & \dots & s[n] \end{bmatrix}^T \tag{II.24}$$

represent a sequence $s[n]$ of length N ending with the observation $s[n]$. In this case the sequences [1]

$$\begin{aligned} \mathbf{s}^e &= \begin{bmatrix} 0 & s[n-N+1] & \dots & 0 & s[n+1] & 0 & s[n] \end{bmatrix}^T \\ \mathbf{s}^o &= \begin{bmatrix} s[n-N+1] & 0 & \dots & s[n+1] & 0 & s[n] & 0 \end{bmatrix}^T \end{aligned} \tag{II.25}$$

[1] Notice that superscripts are used in Equation II.25, while subscripts are used in Equation II.20. The reader should observe that $\mathbf{s}_e \neq \mathbf{s}^e$ and $\mathbf{s}_o \neq \mathbf{s}^o$.

represent upsampling on $s[n]$ up to the n^{th} sample with no time delay or with time delay of 1 accordingly. The two upsampling operations can be represented by the *Upsampling Matrix* $\mathbf{U}^{(0)}$ and *Upsampling Matrix with Time Delay* $\mathbf{U}^{(1)}$ such that:

$$\mathbf{s}^e = \mathbf{U}^{(0)} \cdot \mathbf{s}_n, \quad \mathbf{s}^o = \mathbf{U}^{(1)} \cdot \mathbf{s}_n \tag{II.26}$$

with $\mathbf{U}^{(k)} \in \mathcal{R}^{2N \times N}$. The upsampling and the decimation matrices are related as follows:

$$\mathbf{U}^{(k)} = \mathbf{D}^{(k)T} \quad for \quad k = 0, 1. \tag{II.27}$$

Since upsampling followed by decimation has no effect on the original signal, we can easily show that

$$\mathbf{D}^{(k)} \cdot \mathbf{D}^{(k)T} = \mathbf{I}_N \tag{II.28}$$

where $\mathbf{I}_N$ is the identity matrix of size N. However, decimation followed by upsampling does not restore the original signal, so:

$$\mathbf{U}^{(k)} \cdot \mathbf{U}^{(k)T} \neq \mathbf{I}_{2N}. \tag{II.29}$$

It is also easy to see that the reversal operation changes the unit delay introduced in $\mathbf{D}^{(k)}$, thus:

$$\tilde{\mathbf{D}}^{(0)} = \mathbf{D}^{(1)} \quad and \quad \tilde{\mathbf{D}}^{(1)} = \mathbf{D}^{(0)}. \tag{II.30}$$

C. CONVOLUTION MATRIX

Consider the linear time-invariant operation where a signal $x[n]$ is fed to an FIR filter with impulse response $h[n]$ and of order P. The output $y[n]$ will be the result of the convolution between $x[n]$ and $h[n]$ ($y[n] = x[n] \star h[n]$). Let us represent the sequence $x[n]$ with an observation vector $\mathbf{x}$ of length N. We define the Convolution Matrix of $h[n]$ associated with the vector $\mathbf{x}$ as the following $(P + N - 1) \times N$ matrix:

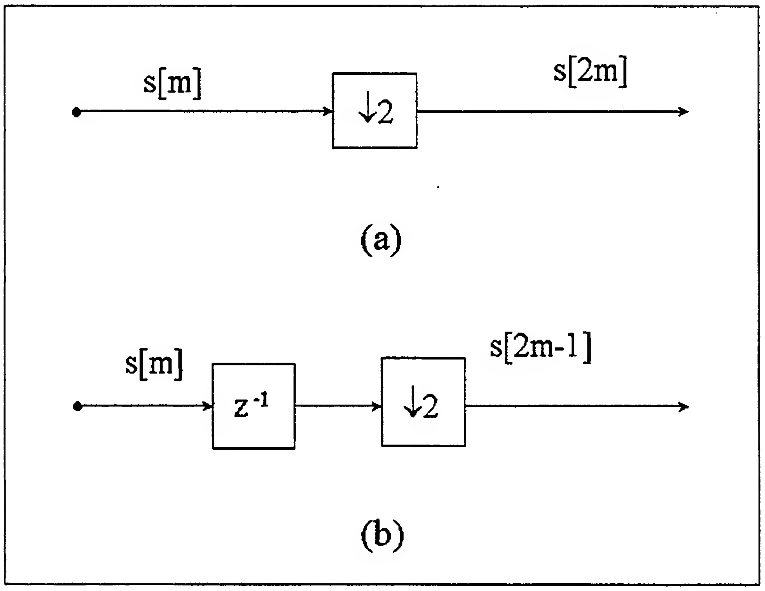

Figure 2. Operations of $\mathbf{D}^{(0)}$ and $\mathbf{D}^{(1)}$

(a) Operation of $\mathbf{D}^{(0)}$ (b) Operation of $\mathbf{D}^{(1)}$

$$\mathbf{H}_N = \begin{bmatrix} h[0] & 0 & \cdots & 0 & 0 \\ h[1] & h[0] & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h[P-1] & h[P-2] & \ddots & 0 & 0 \\ 0 & h[P-1] & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & h[0] & 0 \\ 0 & 0 & \ddots & h[1] & h[0] \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & h[P-1] & h[P-2] \\ 0 & 0 & \cdots & 0 & h[P-1] \end{bmatrix}. \tag{II.31}$$

Then, the output vector of length $N-P+1$ is given by $\mathbf{y} = \mathbf{H}_N \cdot \mathbf{x}$.

The computation of an observation vector may end up significantly simpler with the use of decimation and upsampling matrices as well as convolution matrices. If we assume in particular the system displayed in Figure 3, we can represent the resulting observation vector $\mathbf{x}$ as a matrix product between the vector $\mathbf{s}$ and the corresponding matrices as:

$$\mathbf{x} = \mathbf{D}^{(0)} \cdot \mathbf{H} \cdot \mathbf{s}. \tag{II.32}$$

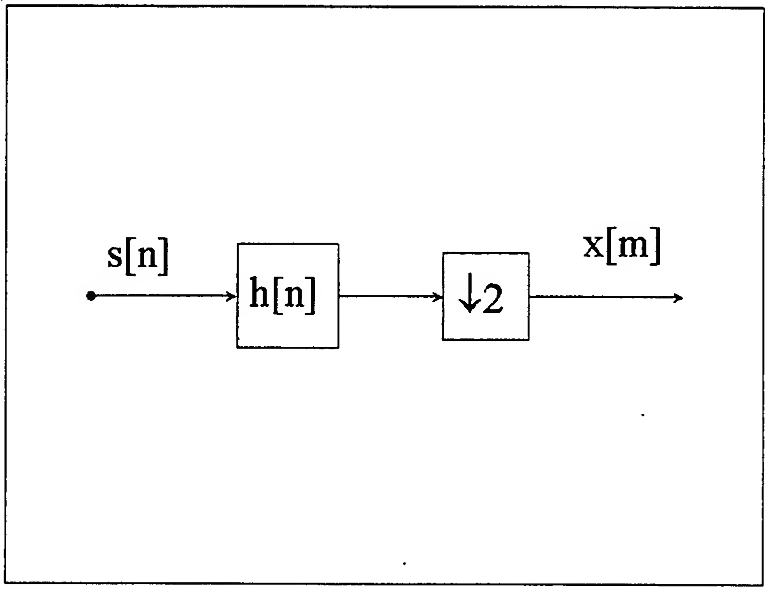

Figure 3. Effect of Decimation and Filtering

D. THE NOBLE IDENTITIES

The *Noble Identities* [Ref. 6: pp. 119-120] represent one set of basic tools for polyphase representation for decimation and interpolation filters. They allow commuting of the decimation or the interpolation operator that leads to simplification of multirate structures. These identities are illustrated in Figure 4. In what follows, we try to form an equivalent of them with the use of vectors and matrices. Let us suppose that we have the finite impulse response of a Linear Time Invariant Filter of order P, $h[n] = \{h[0], h[1], \ldots, h[P-1]\}$. The $\mathcal{Z}$-transform of this filter is denoted

$H(z)$ and we already have the necessary tool for describing the convolution in terms of vectors and matrices from Equation II.31. However, we need a way to describe $H(z^2)$ in order to deal with the Noble Identities. The impulse response that would correspond to $H(z^2)$ would be an *"up sampled version"* of $h[n]$. We can say that $h^{(2)}[n] = \{[h[0], 0, h[1], 0, \ldots, h[P-1], 0\}$, meaning that we have *one non-zero sample every two samples.* In this case, the corresponding convolution matrix is:

$$\mathbf{H}^{(2)} = \underbrace{\begin{bmatrix} h[0] & 0 & \cdots & 0 & 0 \\ 0 & h[0] & \cdots & 0 & 0 \\ h[1] & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h[P-1] & 0 & \ddots & 0 & 0 \\ 0 & h[P-1] & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & h[0] & 0 \\ 0 & 0 & \ddots & 0 & h[0] \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & h[P-1] & 0 \\ 0 & 0 & \cdots & 0 & h[P-1] \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}}_{N} . \qquad \text{(II.33)}$$

The resulting matrix has dimensions $(2P + N - 1) \times N$.

With the introduction of $\mathbf{H}^{(2)}$ we can obtain the *the matrix form of the Noble Identities* that correspond to Figure 4 as:

$$\mathbf{y} = \mathbf{H} \cdot \mathbf{D}^{(0)} \cdot \mathbf{x} \Longleftrightarrow \mathbf{y} = \mathbf{D}^{(0)} \cdot \mathbf{H}^{(2)} \cdot \mathbf{x} \qquad \text{(II.34)}$$

as the first identity, and

$$y = D^{(0)T} \cdot H \cdot x \Longleftrightarrow y = H^{(2)} \cdot D^{(0)T} \cdot x \tag{II.35}$$

as the second identity.

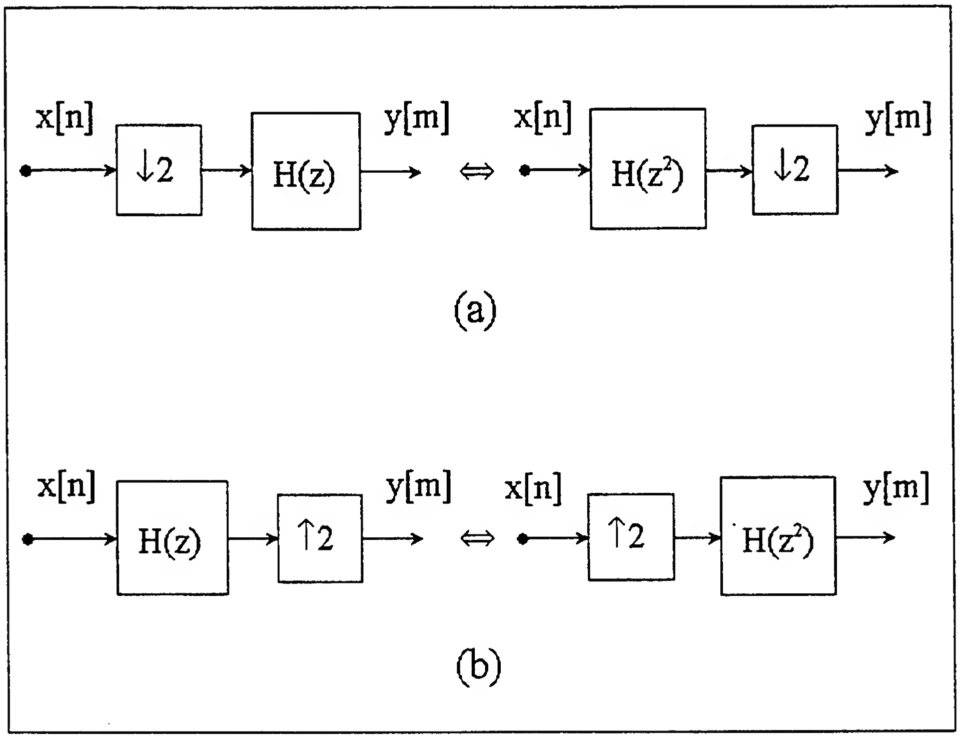

Figure 4. The Noble Identities

(a) Identity No.1 (b) Identity No.2

E. TIME VARYING SYSTEMS AND CYCLOSTATIONARY PROCESSES

In the classical theory of optimal filters we make the assumption that the process to be estimated is WSS and that it is sampled at the same rate as the observed process. As a consequence, optimal filters are also linear time-invariant filters and the stationarity of the estimate is preserved. In the formulation of multirate systems however, this is not the case. Both decimation and upsampling are time-varying operations, and in some cases the output of a multirate system may exhibit periodicity. The simplest example is when a multirate system performs decimation and then upsampling by the same factor, say 2. In this case, if the input signal is

$$x[n] = \{x[0], x[1], x[2], \ldots\} \tag{II.36}$$

then the output signal is

$$y[n] = \{x[0], 0, x[2], 0, x[4], 0, \ldots\}. \tag{II.37}$$

The output $y[n]$ is called a linear periodically time-varying (LPTV) signal [Ref. 6: pp. 130-132]. For the case where the input $x[n]$ is random, wide-sense stationarity is lost in $y[n]$ due to the periodic appearance of the "zeros." In this case, we need to introduce the concept of *cyclostationary* processes [Ref. 9]. A zero mean process $x[n]$ is said to be wide-sense cyclostationary with period M if the correlation function is periodic such that

$$R_x[n+M, k+M] = R_x[n, k]. \tag{II.38}$$

In the example that was previously given, the cross-correlation between the input and the output of the multirate system will be:

$$R_{xy}[n,k] = \begin{cases} R_{xx}[n-k] & , \quad k = 0, 2, 4, \ldots \\ 0 & , \quad k = 1, 3, 5, \ldots \end{cases} \tag{II.39}$$

It turns out that some of the correlation functions that are described in the following section pertain to cyclostationary processes. Furthermore, the period M of the processes is equal to the sampling rate factor.

F. SECOND ORDER ANALYSIS OF LINEAR MULTIRATE SYSTEMS

In the development which follows, we address the effect of multirate systems on the correlation between different signals. Some typical problems are shown in

Figure 5. In all cases under consideration, we wish to determine the cross-correlation matrix between the processes $x[m]$ and $y[n]$ given the fact the $s[n]$ and $y[n]$ are jointly stationary. The fact that we implement all the operations as linear algebraic products simplifies each problem considerably. The results are summarized below.

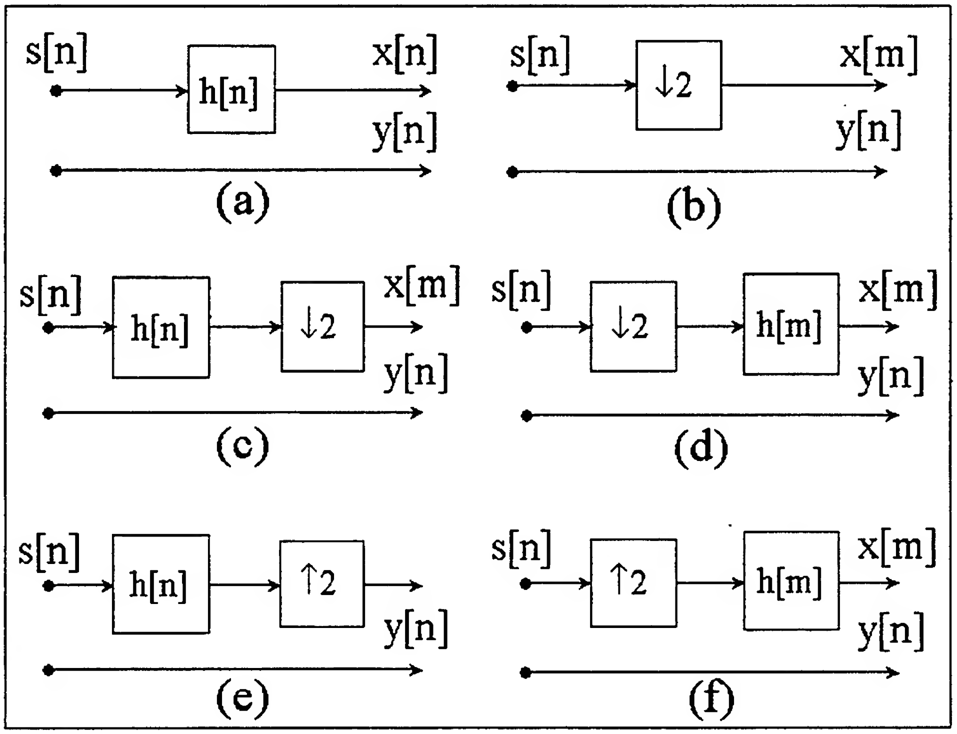

Figure 5. Six Typical Cases of Cross-correlation

(a)	Filtering	(b)	Decimation
(c)	Filtering and Decimation	(d)	Decimation and Filtering
(e)	Filtering and Upsampling	(f)	Upsampling and Filtering

1. Filtering (Figure 5 (a))

When the operation that is performed on $s[n]$ is filtering with a FIR filter $h[n]$ of order P, the resulting cross-correlation matrix can be computed as follows:

$$\mathbf{R}_{xy} = E\left\{\mathbf{x}_n \cdot \mathbf{y}_n^{*T}\right\} = E\left\{\mathbf{H} \cdot \mathbf{s}_n \cdot \mathbf{y}_n^{*T}\right\} = \mathbf{H} \cdot \mathbf{R}_{sy}. \qquad \text{(II.40)}$$

The matrix $\mathbf{R}_{xy}$ has dimensions $(P+N-1)\times N$, the same as $\mathbf{H}$.

2. Decimation (Figure 5 (b))

In the case of decimation by a factor of two, the resulting cross-correlation matrix can be computed as follows:

$$\mathbf{R}_{xy}^{(k)} = E\left\{\mathbf{x}_m \cdot \mathbf{y}_{2m+k}^{*T}\right\} = E\left\{\mathbf{D}^{(k)} \cdot \mathbf{s}_{2m+k} \cdot \mathbf{y}_{2m+k}^{*T}\right\} = \mathbf{D}^{(k)} \cdot \mathbf{R}_{sy}. \tag{II.41}$$

The corresponding cross-correlation function can be computed as follows:

$$R_{xy}^{(k)}[l] = E\left\{x[m] \cdot y^*[2m-2l+k]\right\} = E\left\{s[2m] \cdot y^*[2m-2l+k]\right\} \tag{II.42}$$

which implies

$$R_{xy}^{(k)}[l] = R_{sy}[2l-k]. \tag{II.43}$$

As it was previously stated, the cross-correlation function between $x[m]$ and $y[n]$, represents a cyclostationary process. The index k corresponds to the number of samples that occur within one period. Therefore, for a decimation factor of two, k is 0 or 1.

3. Combination of Filtering and Decimation (Figure 5 (c) & (d))

The last two cases can be combined to use the effects of both filtering and decimation as in (c) and (d) in Figure 5. We have

$$\mathbf{R}_{xy}^{(k)} = E\left\{\mathbf{x}_m \cdot \mathbf{y}_{2m+k}^{*T}\right\} = E\left\{\mathbf{D}^{(k)} \cdot \mathbf{H} \cdot \mathbf{s}_{2m+k} \cdot \mathbf{y}_{2m+k}^{*T}\right\} \tag{II.44}$$

leading to

$$\mathbf{R}_{xy}^{(k)} = \mathbf{D}^{(k)} \cdot \mathbf{H} \cdot \mathbf{R}_{sy}. \tag{II.45}$$

In the case of decimation prior to filtering, we can write

$$\mathbf{R}_{xy}^{(k)} = E\left\{\mathbf{x}_m \cdot \mathbf{y}_{2m+k}^{*T}\right\} = E\left\{\mathbf{H} \cdot \mathbf{D}^{(k)} \cdot \mathbf{s}_{2m+k} \cdot \mathbf{y}_{2m+k}^{*T}\right\} \tag{II.46}$$

and finally

$$\mathbf{R}_{xy}^{(k)} = \mathbf{H} \cdot \mathbf{D}^{(k)} \cdot \mathbf{R}_{sy}. \tag{II.47}$$

4. Combination of Filtering and Upsampling (Figure 5 (e) & (f))

A similar case may occur when, instead of decimation we perform upsampling on the input signal. In the case where upsampling is performed prior to filtering, the resulting cross-correlation matrix is computed as:

$$\mathbf{R}_{xy}^{(k)} = E\left\{\mathbf{x}_m \cdot \mathbf{y}_n^{*T}\right\} = E\left\{\mathbf{H} \cdot \mathbf{D}^{(k)T} \cdot \mathbf{s}_n \cdot \mathbf{y}_n^{*T}\right\} \tag{II.48}$$

resulting in

$$\mathbf{R}_{xy}^{(k)} = \mathbf{H} \cdot \mathbf{D}^{(k)T} \cdot \mathbf{R}_{sy}. \tag{II.49}$$

When filtering is performed prior to upsampling, the resulting cross-correlation matrix becomes:

$$\mathbf{R}_{xy}^{(k)} = E\left\{\mathbf{x}_m \cdot \mathbf{y}_n^{*T}\right\} = E\left\{\mathbf{D}^{(k)T} \cdot \mathbf{H} \cdot \mathbf{s}_n \cdot \mathbf{y}_n^{*T}\right\} \tag{II.50}$$

which implies

$$\mathbf{R}_{xy}^{(k)} = \mathbf{D}^{(k)T} \cdot \mathbf{H} \cdot \mathbf{R}_{sy}. \tag{II.51}$$

A summary of the results from Equations II.40 through II.51 is provided in Table II. These results are fundamental to the derivation of an optimal filter that implements two sets of observations at different sampling rates.

Operation	Result
Filtering	$\mathbf{R}_{xy} = \mathbf{H} \cdot \mathbf{R}_{sy}$
Decimation	$\mathbf{R}_{xy}^{(k)} = \mathbf{D}^{(k)} \cdot \mathbf{R}_{sy}$
Filtering and Decimation	$\mathbf{R}_{xy}^{(k)} = \mathbf{D}^{(k)} \cdot \mathbf{H} \cdot \mathbf{R}_{sy}$
Decimation and Filtering	$\mathbf{R}_{xy}^{(k)} = \mathbf{H} \cdot \mathbf{D}^{(k)} \cdot \mathbf{R}_{sy}$
Upsampling and Filtering	$\mathbf{R}_{xy}^{(k)} = \mathbf{H} \cdot \mathbf{D}^{(k)T} \cdot \mathbf{R}_{sy}$
Filtering and Upsampling	$\mathbf{R}_{xy}^{(k)} = \mathbf{D}^{(k)T} \cdot \mathbf{H} \cdot \mathbf{R}_{sy}$

Table II. Cross-correlation Table

THIS PAGE INTENTIONALLY LEFT BLANK

III. THE MULTIRATE WIENER FILTER

One major class of applications in multirate digital signal processing are the information fusion problems. The general concept in this class of problems is the following: Based on the combination of observations available from different sources, obtain the best possible estimate on a signal, to which the observations are somehow related.

In accordance with the principle that was previously stated, we form the problem that is illustrated in Figure 6. We wish to estimate a signal $s[n]$, from which we can obtain two observations $x[n]$ and $y[m]$, related to the original signal, where the different indices denote different sampling rates. In particular, both signals $s[n]$ and $x[n]$ are sampled at the highest possible rate, while $y[m]$ is sampled at a lower rate; specifically the sampling rate for $y[m]$ comes from decimation by an integer factor. We want to compute the estimate $\hat{s}[n]$ at the same rate as that of the original signal.

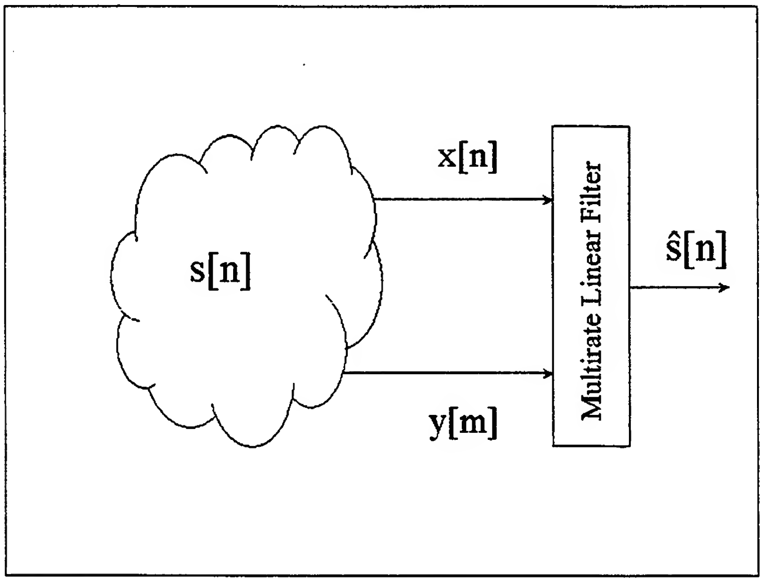

Figure 6. Basic Multirate Optimal Filtering Problem

We consider the specific case where $y[m]$ is at half the sampling rate of $s[n]$ (decimation by a factor of two), and $s[n]$ is zero-mean and wide sense stationary (WSS). However the problem can be easily extended to more than two observation sequences, each obtained with the use of a different decimation factor from the sampling rate of the original signal.

We shall see that the estimate can be implemented as shown in Figure 7. The optimal filter consists of two branches that take as an input the observation signals $x[n]$ and $y[m]$. After filtering from the corresponding functions $\mathbf{h}^{(k)}$ and $\mathbf{g}^{(k)}$, the high sampling rate output is decimated by a factor of two. Both outputs are added in order to provide the even and odd samples of the estimated signal $\hat{s}[n]$. Finally both samples are clocked through a parallel-to-serial converter to produce the estimate. This structure is extendable to higher decimation factors. Thus, if the sampling rates of the observations were related to a decimation factor of three, the input signals to the parallel-to-serial stage would simply be three as well ($\hat{s}[3m+k]$, where $k = 0, 1, 2$).

A. THE MULTIRATE WIENER-HOPF EQUATION

From the fact that we have two observation sequences available, we form two observation vectors which we call $\mathbf{x}_{2m+k}$ with N elements and $\mathbf{y}_m$ with M elements. Since $y[m]$ is sampled at half the rate, it is convenient to use the expression $2m + k$, where $k = 0, 1$ to represent the index of $x[n]$. The observation vector that is formed by combining $\mathbf{x}_{2m+k}$ and $\mathbf{y}_m$ has the form:

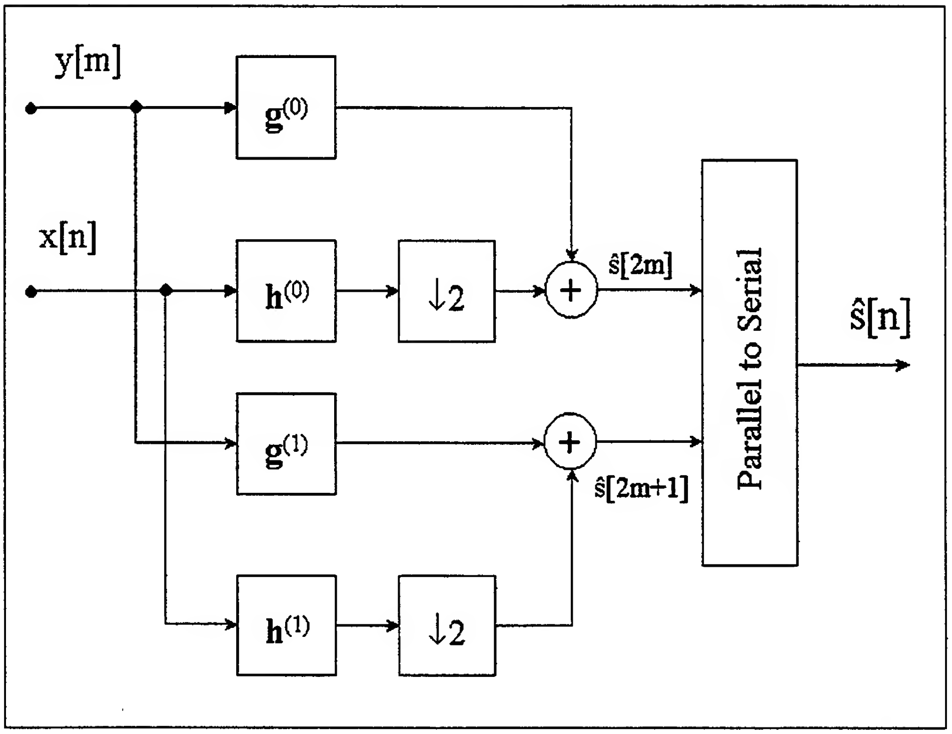

Figure 7. Implementation of Multirate Wiener Filter

$$\begin{bmatrix} \mathbf{x}_{2m+k} \\ -- \\ \mathbf{y}_m \end{bmatrix} = \begin{bmatrix} x[2m-N+1+k] \\ x[2m-N+2+k] \\ \vdots \\ x[2m+k] \\ --- \\ y[m-M+1] \\ y[m-M+2] \\ \vdots \\ y[m] \end{bmatrix}. \tag{III.1}$$

We start from the fact that the optimal mean-squared filter satisfies the orthogonality principle or, in other words, the error from the obtained estimate is orthogonal to the observation vectors. Define $\varepsilon[n]$ as the error of the estimation, $\varepsilon[n] = s[n] - \hat{s}[n]$. The orthogonality principle dictates that:

$$E\left\{\left[\begin{array}{c}\tilde{\mathbf{x}}_{2m+k}\\ \tilde{\mathbf{y}}_m\end{array}\right]\cdot \varepsilon^*[n]\right\} = \mathbf{0} \tag{III.2}$$

where $k = 0, 1$.

We want to express the estimate $\hat{s}[n]$ as the result of linear filtering of the observation vector from a filter that obviously is not shift invariant. In addition, we can speculate that this filter is divided into two distinct filtering functions $h[n, m]$ (for $x[n]$) and $g[n, m]$ (for $y[m]$) and that they also depend on the index k due to decimation. We can write:

$$\hat{s}[2m+k] = \sum_{l=0}^{N-1} h^{(k)}[2m+k, 2m+k-l]\cdot x[2m+k-l] + \sum_{\lambda=0}^{M-1} g^{(k)}[m, m-\lambda]\cdot y[m-\lambda]. \tag{III.3}$$

In order to make the notation more compact, let us express the impulse responses of both filters as vectors. The estimate then becomes:

$$\hat{s}[2m+k] = \left[\begin{array}{c|c}\tilde{\mathbf{x}}_{2m+k}^T & \tilde{\mathbf{y}}_m^T\end{array}\right]\cdot\left[\begin{array}{c}\mathbf{h}^{(k)}\\ \mathbf{g}^{(k)}\end{array}\right] \tag{III.4}$$

where $k = 0, 1$ or

$$\hat{s}[2m+k] = \tilde{\mathbf{x}}_{2m+k}^T \cdot \mathbf{h}^{(k)} + \tilde{\mathbf{y}}_m^T \cdot \mathbf{g}^{(k)}. \tag{III.5}$$

Since reversing both terms in an inner product does not change the result, we can also use the following alternative expression:

$$\hat{s}[2m+k] = \mathbf{h}^{(k)T}\cdot \tilde{\mathbf{x}}_{2m+k} + \mathbf{g}^{(k)T}\cdot \tilde{\mathbf{y}}_m. \tag{III.6}$$

With the last expression in hand, we can now form the relations that satisfy the Orthogonality Principle. Equation III.2 becomes:

$$E\left\{\begin{bmatrix}\tilde{\mathbf{x}}_{2m+k}\\ \tilde{\mathbf{y}}_m\end{bmatrix}\cdot\left(s^*[2m+k]-\hat{s}^*[2m+k]\right)\right\}=\mathbf{0} \tag{III.7}$$

and by substituting Equation III.6 we obtain

$$E\left\{\begin{bmatrix}\tilde{\mathbf{x}}_{2m+k}\\ \tilde{\mathbf{y}}_m\end{bmatrix}\cdot\left(s^*[2m+k]-\begin{bmatrix}\tilde{\mathbf{x}}_{2m+k}\\ \tilde{\mathbf{y}}_m\end{bmatrix}^{*T}\cdot\begin{bmatrix}\mathbf{h}^{(k)*}\\ \mathbf{g}^{(k)*}\end{bmatrix}\right)\right\}=\mathbf{0}. \tag{III.8}$$

By making the necessary operations,

$$E\left\{\begin{bmatrix}\tilde{\mathbf{x}}_{2m+k}\\ \tilde{\mathbf{y}}_m\end{bmatrix}\cdot s^*[2m+k]\right\}=E\left\{\begin{bmatrix}\tilde{\mathbf{x}}_{2m+k}\\ \tilde{\mathbf{y}}_m\end{bmatrix}\cdot\begin{bmatrix}\tilde{\mathbf{x}}_{2m+k}\\ \tilde{\mathbf{y}}_m\end{bmatrix}^{*T}\cdot\begin{bmatrix}\mathbf{h}^{(k)*}\\ \mathbf{g}^{(k)*}\end{bmatrix}\right\} \tag{III.9}$$

we end up with

$$\left[\begin{array}{c|c}\mathbf{R}^*_{xx} & \mathbf{R}^{(k)*}_{xy}\\ & \\ (\mathbf{R}^{(k)*T}_{xy})^* & \mathbf{R}^*_{yy}\end{array}\right]\cdot\begin{bmatrix}\mathbf{h}^{(k)*}\\ \\ \mathbf{g}^{(k)*}\end{bmatrix}=\begin{bmatrix}\tilde{\mathbf{r}}^*_{sx}\\ \\ \tilde{\mathbf{r}}^{(k)*}_{sy}\end{bmatrix} \tag{III.10}$$

where we have defined:

$$\mathbf{R}^*_{xx}=E\left\{\tilde{\mathbf{x}}_{2m+k}\cdot\tilde{\mathbf{x}}^{*T}_{2m+k}\right\} \tag{III.11}$$

$$\mathbf{R}^*_{yy}=E\left\{\tilde{\mathbf{y}}_m\cdot\tilde{\mathbf{y}}^{*T}_m\right\} \tag{III.12}$$

$$\mathbf{R}^{(k)*}_{xy}=E\left\{\tilde{\mathbf{x}}_{2m+k}\cdot\tilde{\mathbf{y}}^{*T}_m\right\} \tag{III.13}$$

$$\mathbf{r}_{sx} = E\left\{s[2m+k] \cdot \tilde{\mathbf{x}}^*_{2m+k}\right\} \tag{III.14}$$

$$\mathbf{r}^{(k)}_{sy} = E\left\{s[2m+k] \cdot \tilde{\mathbf{y}}^*_m\right\}. \tag{III.15}$$

Note that $\mathbf{R}^{(k)}_{xy}$ and $\mathbf{r}^{(k)}_{sy}$ are functions of k, while all other terms are not. Finally, by conjugating all terms in Equation III.10, we obtain:

$$\left[\begin{array}{c|c} \mathbf{R}_{xx} & \mathbf{R}^{(k)}_{xy} \\ & \\ \mathbf{R}^{(k)*T}_{xy} & \mathbf{R}_{yy} \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{h}^{(k)} \\ \\ \mathbf{g}^{(k)} \end{array}\right] = \left[\begin{array}{c} \tilde{\mathbf{r}}_{sx} \\ \\ \tilde{\mathbf{r}}^{(k)}_{sy} \end{array}\right] \tag{III.16}$$

for $k = 0, 1$.

We refer to Equation III.16 as the *multirate Wiener-Hopf equation.*

The filter that processes the estimate for $s[n]$ is *linear* but *time-varying.* The vector that comes from the permutation of $\mathbf{h}^{(k)}$ and $\mathbf{g}^{(k)}$ has length $M + N$, and since it is derived from the orthogonality principle, it produces the Minimum Mean Square Error (MMSE) estimate, given observations from the signals $x[n]$ and $y[m]$. The estimate is given by Equation III.6 and implemented as the combination of two periodically time-varying linear filters, with a period equal to the decimation factor (2). To express the estimate in scalar form, we can write:

$$\hat{s}[2m+k] = \sum_{l=0}^{N-1} h^{(k)}[l] \cdot x[2m+k-l] + \sum_{\lambda=0}^{M-1} g^{(k)}[\lambda] \cdot y[m-\lambda] \tag{III.17}$$

which is identical to Equation III.3, except we have observed that the solution to Equation III.16 does not depend on m, and have thus written the filter terms as $h^{(k)}[l]$ and $g^{(k)}[\lambda]$. The optimal filter is then seen to have the implementation given in Figure 7.

We can use the fact that there is a solution to the filter to express the minimum error variance, as:

$$\sigma_\varepsilon^2(k) = E\left\{s[2m+k]\cdot\varepsilon^*[2m+k]\right\} \quad , \quad k = 0,1. \qquad \text{(III.18)}$$

Note that the minimum mean square error is a function of the index k due to the periodically time-varying nature of the filter. By substituting the expression for $\varepsilon^*[2m+k]$, we obtain:

$$\begin{aligned}
\sigma_\varepsilon^2(k) &= E\left\{s[2m+k]\cdot\left(s[2m+k]-\left[\begin{array}{c|c}\tilde{\mathbf{x}}_{2m+k}^T & \tilde{\mathbf{y}}_m^T\end{array}\right]\cdot\left[\begin{array}{c}\mathbf{h}^{(k)}\\ \mathbf{g}^{(k)}\end{array}\right]\right)^*\right\}\\
&= E\left\{s[2m+k]\cdot s^*[2m+k]\right\} - E\left\{s[2m+k]\cdot\left[\begin{array}{c|c}\tilde{\mathbf{x}}_{2m+k}^{*T} & \tilde{\mathbf{y}}_m^{*T}\end{array}\right]\cdot\left[\begin{array}{c}\mathbf{h}^{(k)*}\\ \mathbf{g}^{(k)*}\end{array}\right]\right\}\\
&= R_{ss}[0]-\left[\begin{array}{c|c}\tilde{\mathbf{r}}_{sx}^T & \tilde{\mathbf{r}}_{sy}^{(k)T}\end{array}\right]\cdot\left[\begin{array}{c}\mathbf{h}^{(k)*}\\ \mathbf{g}^{(k)*}\end{array}\right]\\
&= R_{ss}[0]-\tilde{\mathbf{r}}_{sx}^T\cdot\mathbf{h}^{(k)*}-\tilde{\mathbf{r}}_{sy}^{(k)T}\cdot\mathbf{g}^{(k)*}
\end{aligned} \qquad \text{(III.19)}$$

or:

$$\sigma_\varepsilon^2(k) = R_{ss}[0]-\mathbf{h}^{(k)*T}\cdot\tilde{\mathbf{r}}_{sx}-\mathbf{g}^{(k)*T}\cdot\tilde{\mathbf{r}}_{sy}^{(k)}. \qquad \text{(III.20)}$$

B. COMPARISON OF SINGLE RATE AND MULTIRATE WIENER FILTERS

We already have a form for the σ_ε^2, the MMSE that is the result of the multirate Wiener filter. It was previously stated that the derivation of the filter was based on the same principles as the single rate Wiener filter. Therefore, it is only natural to pursue a quantitative comparison between the two. First we will compare the expression for the filter in these two cases and provide a connection between them. Secondly, if we denote as σ_0^2 the MMSE that is the result of the use of the single rate Wiener filter on only the high sampling rate observations $x[n]$, we would like to show that σ_ε^2 (the MMSE from the multirate filter) is always a smaller quantity than σ_0^2 and be able to derive an expression for the difference.

To begin, let us consider the following:

Lemma 1 *Consider a square matrix K of the form*

$$\mathbf{K} = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^{*T} & \mathbf{C} \end{array}\right]. \tag{III.21}$$

Then the inverse $\mathbf{K}^{-1}$ can be expressed as

$$\mathbf{K}^{-1} = \left[\begin{array}{c|c} \mathbf{A}^{-1} + \mathbf{G}\cdot\mathbf{E}^{-1}\cdot\mathbf{G}^{*T} & -\mathbf{G}\cdot\mathbf{E}^{-1} \\ -\mathbf{E}^{-1}\cdot\mathbf{G}^{*T} & \mathbf{E}^{-1} \end{array}\right] \tag{III.22}$$

where

$$\begin{aligned} \mathbf{G} &= \mathbf{A}^{-1}\cdot\mathbf{B} \\ \mathbf{E} &= \mathbf{C} - \mathbf{B}^{*T}\cdot\mathbf{A}^{-1}\cdot\mathbf{B} \end{aligned} \tag{III.23}$$

and where all of the cited inverse matrices are assumed to exist.

Applying the results of Lemma 1 to the correlation matrix $\mathbf{R}^{(k)}$, defined in left-hand side of Equation III.16, we can write

$$\mathbf{R}^{(k)^{-1}} = \left[\begin{array}{c|c} \mathbf{R}_{xx}{}^{-1} + \mathbf{G}_k\cdot\mathbf{E}_k{}^{-1}\cdot\mathbf{G}_k^{*T} & -\mathbf{G}_k\cdot\mathbf{E}_k{}^{-1} \\ -\mathbf{E}_k{}^{-1}\cdot\mathbf{G}_k^{*T} & \mathbf{E}_k{}^{-1} \end{array}\right] \tag{III.24}$$

where

$$\begin{aligned} \mathbf{G}_k &= \mathbf{R}_{xx}{}^{-1}\cdot\mathbf{R}_{xy}^{(k)} \\ \mathbf{E}_k &= \mathbf{R}_{yy} - \mathbf{R}_{xy}^{(k)*T}\cdot\mathbf{R}_{xx}{}^{-1}\cdot\mathbf{R}_{xy}^{(k)}. \end{aligned} \tag{III.25}$$

Substituting in Equation III.24 to express the filter coefficients $\mathbf{h}^{(k)}$ and $\mathbf{g}^{(k)}$, we obtain:

$$\begin{bmatrix} \mathbf{h}^{(k)} \\ \\ \mathbf{g}^{(k)} \end{bmatrix} = \left\{ \left[\begin{array}{c|c} \mathbf{R}_{xx}{}^{-1} & \mathbf{0} \\ & \\ \mathbf{0} & \mathbf{0} \end{array} \right] + \begin{bmatrix} -\mathbf{G}_k \\ \\ \mathbf{I} \end{bmatrix} \cdot \mathbf{E}_k{}^{-1} \cdot \left[\begin{array}{c|c} -\mathbf{G}_k^{*T} & \mathbf{I} \end{array} \right] \right\} \cdot \begin{bmatrix} \tilde{\mathbf{r}}_{sx} \\ \\ \tilde{\mathbf{r}}_{sy}^{(k)} \end{bmatrix} \tag{III.26}$$

where $\mathbf{0}$ denotes an all-zero matrix with appropriate dimensions and $\mathbf{I}$ denotes the identity matrix. In order to assure the existence of $\mathbf{h}^{(k)}$ and $\mathbf{g}^{(k)}$, the matrix $\mathbf{E}_k$ must be invertible. We know that $\mathbf{E}_k$ is a submatrix that shares the main diagonal with $\mathbf{R}^{(k)}$. In all cases of interest, where noise is present, $\mathbf{R}^{(k)}$ is invertible. Since $\mathbf{R}^{(k)}$ is a positive definite matrix, $\mathbf{E}_k$ is also invertible and positive definite [Ref. 10: pp.333]. Therefore, we can express $\mathbf{h}^{(k)}$ and $\mathbf{g}^{(k)}$ as:

$$\mathbf{h}^{(k)} = \mathbf{R}_{xx}{}^{-1} \cdot \tilde{\mathbf{r}}_{sx} - \mathbf{G}_k \cdot \mathbf{E}_k{}^{-1} \cdot \left(\tilde{\mathbf{r}}_{sy}^{(k)} - \mathbf{G}_k^{*T} \cdot \tilde{\mathbf{r}}_{sx} \right) \tag{III.27}$$

$$\mathbf{g}^{(k)} = \mathbf{E}_k{}^{-1} \cdot \left(\tilde{\mathbf{r}}_{sy}^{(k)} - \mathbf{G}_k^{*T} \cdot \tilde{\mathbf{r}}_{sx} \right). \tag{III.28}$$

The term $\mathbf{h}_0 = \mathbf{R}_{xx}{}^{-1} \cdot \tilde{\mathbf{r}}_{sx}$ is the well known solution of the single rate Wiener-Hopf Equation (see Equation II.2). Therefore, by combining Equations III.27 and III.28, we obtain:

$$\mathbf{h}^{(k)} = \mathbf{h}_0 - \mathbf{G}_k \cdot \mathbf{g}^{(k)}. \tag{III.29}$$

Equations III.27 through III.29 establish a relationship between the single rate Wiener filter and its multirate extension. This is particularly significant when we compute the mean squared error of the estimates to compare the two implementations, single rate and multirate.

The next step is to compute a new formula for σ_ε^2 based on the previously stated equations. From Equation III.6, by substituting for the filter coefficients, we can write:

$$\begin{aligned}
\hat{s}[2m+k] &= \tilde{\mathbf{x}}_{2m+k}^T \cdot \mathbf{h}^{(k)} + \tilde{\mathbf{y}}_m^T \cdot \mathbf{g}^{(k)} \\
&= \tilde{\mathbf{x}}_{2m+k}^T \cdot (\mathbf{h}_0 - \mathbf{G}_k \cdot \mathbf{g}^{(k)}) + \tilde{\mathbf{y}}_m^T \cdot \mathbf{g}^{(k)} \\
&= \tilde{\mathbf{x}}_{2m+k}^T \cdot \mathbf{h}_0 - \tilde{\mathbf{x}}_{2m+k}^T \cdot \mathbf{G}_k \cdot \mathbf{g}^{(k)} + \tilde{\mathbf{y}}_m^T \cdot \mathbf{g}^{(k)} \\
&= \tilde{\mathbf{x}}_{2m+k}^T \cdot \mathbf{h}_0 + (\tilde{\mathbf{y}}_m^T - \tilde{\mathbf{x}}_{2m+k}^T \cdot \mathbf{G}_k) \cdot \mathbf{g}^{(k)}
\end{aligned} \tag{III.30}$$

which implies

$$\hat{s}[2m+k] = \hat{s}_0[2m+k] + (\tilde{\mathbf{y}}_m^T - \tilde{\mathbf{x}}_{2m+k}^T \cdot \mathbf{G}_k) \cdot \mathbf{g}^{(k)}. \tag{III.31}$$

where $\hat{s}_0[n]$ is the estimate from the Wiener filter based on the solution of the single rate Wiener-Hopf equations for $x[n]$ as the only observation. We can see that the final estimate is the sum of $\hat{s}_0[n]$ plus a correction due to the new observations $y[m]$.

Now let us consider the mean-square error expression and show that the multirate estimation gives a better estimate (in the mean squared sense) than the single rate Wiener filter with only one observation. By substituting Equation III.29 into Equation III.20, we obtain:

$$\begin{aligned}
\sigma_\varepsilon^2(k) &= R_{ss}[0] - \mathbf{h}^{(k)*T} \cdot \tilde{\mathbf{r}}_{sx} - \mathbf{g}^{(k)*T} \cdot \tilde{\mathbf{r}}_{sy}^{(k)} \\
&= R_{ss}[0] - \left(\mathbf{h}_0 - \mathbf{G}_k \cdot \mathbf{g}^{(k)}\right)^{*T} \tilde{\mathbf{r}}_{sx} - \mathbf{g}^{(k)*T} \cdot \tilde{\mathbf{r}}_{sy}^{(k)} \\
&= \underbrace{R_{ss}[0] - \mathbf{h}_0^{*T} \cdot \tilde{\mathbf{r}}_{sx}}_{\sigma_0^2} + \mathbf{g}^{(k)*T} \cdot \mathbf{G}_k^{*T} \cdot \tilde{\mathbf{r}}_{sx} - \mathbf{g}^{(k)*T} \cdot \tilde{\mathbf{r}}_{sy}^{(k)}
\end{aligned} \tag{III.32}$$

or

$$\sigma_\varepsilon^2(k) = \sigma_0^2 - \mathbf{g}^{(k)*T} \cdot \left(\tilde{\mathbf{r}}_{sy}^{(k)} - \mathbf{G}_k^{*T} \cdot \tilde{\mathbf{r}}_{sx}\right) \tag{III.33}$$

where $\sigma_0^2 = R_{ss}[0] - \mathbf{h}_0^{*T} \cdot \tilde{\mathbf{r}}_{sx}$ is the error covariance of the single rate Wiener filter. We can show that the second term in the right-hand side of Equation III.33 is always positive. Using Equation III.28, we obtain:

$$\begin{aligned} \mathbf{g}^{(k)*T} \cdot \left(\tilde{\mathbf{r}}_{sy}^{(k)} - \mathbf{G}_k^{*T} \cdot \tilde{\mathbf{r}}_{sx}\right) &= \left(\mathbf{E}_k{}^{-1} \left(\tilde{\mathbf{r}}_{sy}^{(k)} - \mathbf{G}_k^{*T} \cdot \tilde{\mathbf{r}}_{sx}\right)\right)^{*T} \cdot \left(\tilde{\mathbf{r}}_{sy}^{(k)} - \mathbf{G}_k^{*T} \cdot \tilde{\mathbf{r}}_{sx}\right) \\ &= \left(\tilde{\mathbf{r}}_{sy}^{(k)} - \mathbf{G}_k^{*T} \cdot \tilde{\mathbf{r}}_{sx}\right)^{*T} \cdot \mathbf{E}_k{}^{-1} \cdot \left(\tilde{\mathbf{r}}_{sy}^{(k)} - \mathbf{G}_k^{*T} \cdot \tilde{\mathbf{r}}_{sx}\right) \end{aligned} \tag{III.34}$$

where we have used the fact that $\mathbf{E}_k$ is Hermitian symmetric, so $\mathbf{E}_k^{*T} = \mathbf{E}_k$. Since $\mathbf{E}_k$ is positive definite, the last expression is a *scalar quadratic form of a positive definite matrix*; therefore it is always positive. Since $\sigma_\varepsilon^2(k)$ and σ_0^2 are both positive, we can conclude from Equation III.33 that

$$\sigma_\varepsilon^2 < \sigma_0^2. \tag{III.35}$$

That is, the MSE of the multirate Wiener filter is always smaller than the MSE of the single rate Wiener filter.

C. ESTIMATION OF FILTERED SIGNALS IN NOISE

Based on the framework set in the previous section, we can address the following problem. A random signal $s[n]$ which is zero-mean and WSS, is observed through two distinct channels at different sampling rates. We assume that a decimation by a factor of 2 takes place in one of the channels, while both observations are corrupted by noise of known covariance, independent on the signal $s[n]$. This is shown in Figure 8, where the observations are denoted by $x[n]$ and $y[m]$, and the noise sequences by $u[n]$ and $v[m]$ respectively. The goal is to compute an estimate for $s[n]$, called $\hat{s}[n]$, that is the minimum mean-squared estimate for the desired signal, based on the two sequences of observations. The FIR filter that provides this estimate is called the multirate Wiener filter for the problem.

In order to make the arguments easier to follow, we first consider a simplified case in which there are no transformation filters or, for both filtering functions in Figure 8, $\Theta(z) = \Gamma(z) = 1$. The last assumption makes the derivation of the filter less tedious. Following that original problem, we consider the generalization of the previous case with impulse responses $\theta[n]$ and $\gamma[n]$ of orders P and Q respectively.

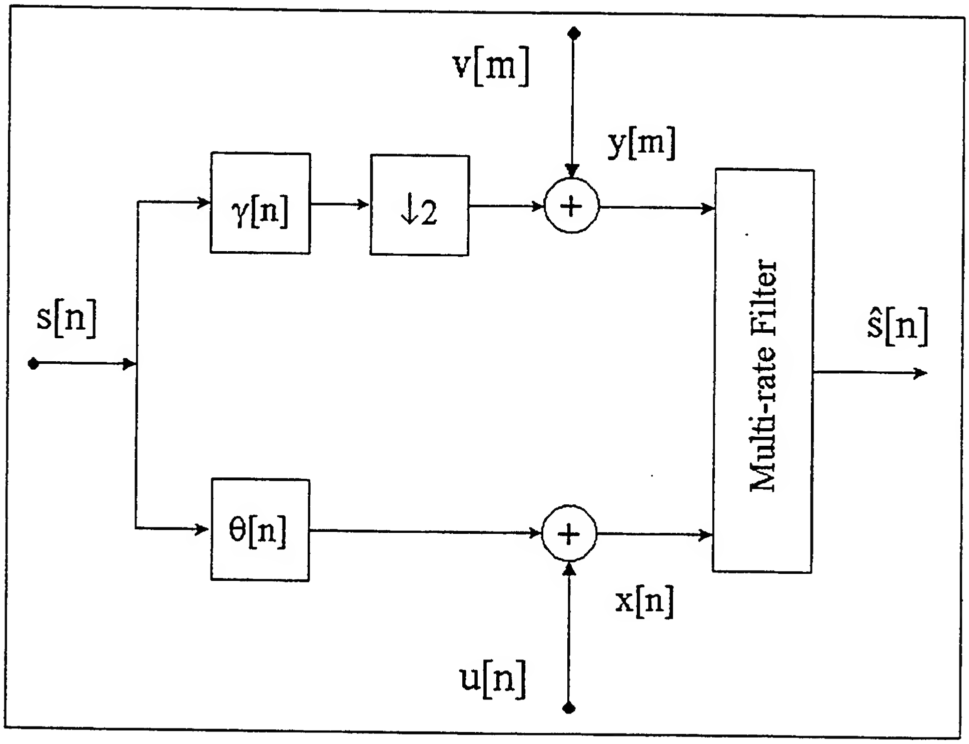

Figure 8. Block Diagram of Multirate Wiener Filter

1. Unfiltered Observations

We start with the simplified case, for which $\Theta(z) = \Gamma(z) = 1$ and assume that $u[n]$ and $v[m]$ as white noise sequences of variance σ_u^2 and σ_v^2 respectively. Based on the problem parameters, we can compute each of the elements in both sides of the Wiener-Hopf equations for the multirate Wiener filter, as stated in Equation III.16. The resulting equations involve considerable simplifications, since both $u[n]$ and $v[m]$ are white noise processes, uncorrelated with any other signal. The correlation matrix for the high sampling rate signal $x[n]$ is:

$$\begin{aligned}
\mathbf{R}_{xx}^{*} &= E\left\{\tilde{\mathbf{x}}_{2m+k} \cdot \tilde{\mathbf{x}}_{2m+k}^{*T}\right\} = E\left\{(\tilde{\mathbf{s}}_{2m+k} + \tilde{\mathbf{u}}_{2m+k}) \cdot (\tilde{\mathbf{s}}_{2m+k} + \tilde{\mathbf{u}}_{2m+k})^{*T}\right\} \\
&= E\left\{\tilde{\mathbf{s}}_{2m+k} \cdot \tilde{\mathbf{s}}_{2m+k}^{*T}\right\} + E\left\{\tilde{\mathbf{s}}_{2m+k} \cdot \tilde{\mathbf{u}}_{2m+k}^{*T}\right\} \\
&\quad + E\left\{\tilde{\mathbf{u}}_{2m+k} \cdot \tilde{\mathbf{s}}_{2m+k}^{*T}\right\} + E\left\{\tilde{\mathbf{u}}_{2m+k} \cdot \tilde{\mathbf{u}}_{2m+k}^{*T}\right\} \qquad \text{(III.36)} \\
&= \tilde{\mathbf{R}}_{ss}^{(N,N)} + \sigma_u^2 \cdot \mathbf{I}_N
\end{aligned}$$

leading to

$$\mathbf{R}_{xx} = \mathbf{R}_{ss}^{(N,N)} + \sigma_u^2 \cdot \mathbf{I}_N \qquad \text{(III.37)}$$

where the cross terms are zero and we have used the identity $\tilde{\mathbf{R}}_{ss} = \mathbf{R}_{ss}^{*}$ (see Equation II.13), since the signal is stationary in the wide sense. The superscript (N, N) in the correlation matrix ($\mathbf{R}_{ss}^{(N,N)}$) represents the Toeplitz matrix produced from $R_{ss}[l]$ and has N rows and N columns. However, the number of rows may not be necessarily the same with the number of columns, as it will be stated in the following derivations. The correlation matrix of the low sampling rate signal $y[m]$ becomes:

$$\begin{aligned}
\mathbf{R}_{yy}^{*} &= E\left\{\tilde{\mathbf{y}}_m \cdot \tilde{\mathbf{y}}_m^{*T}\right\} = E\left\{\left(\tilde{\mathbf{D}}^{(0)} \cdot \tilde{\mathbf{s}}_{2m+k} + \tilde{\mathbf{v}}_m\right) \cdot \left(\tilde{\mathbf{D}}^{(0)} \cdot \tilde{\mathbf{s}}_{2m+k} + \tilde{\mathbf{v}}_m\right)^{*T}\right\} \\
&= E\left\{\mathbf{D}^{(1)} \cdot \tilde{\mathbf{s}}_{2m+k} \cdot \tilde{\mathbf{s}}_{2m+k}^{*T} \cdot \mathbf{D}^{(1)T}\right\} + E\left\{\mathbf{D}^{(1)} \cdot \tilde{\mathbf{s}}_{2m+k} \cdot \tilde{\mathbf{v}}_m^{*T}\right\} \\
&\quad + E\left\{\tilde{\mathbf{v}}_m \cdot \tilde{\mathbf{s}}_{2m+k}^{*T} \cdot \mathbf{D}^{(1)T}\right\} + E\left\{\tilde{\mathbf{v}}_m \cdot \tilde{\mathbf{v}}_m^{*T}\right\} \\
&= \mathbf{D}^{(1)} \cdot \tilde{\mathbf{R}}_{ss}^{(2M,2M)} \cdot \mathbf{D}^{(1)T} + \sigma_v^2 \cdot \mathbf{I}_M
\end{aligned} \qquad \text{(III.38)}$$

which yields

$$\mathbf{R}_{yy} = \mathbf{D}^{(1)} \cdot \mathbf{R}_{ss}^{(2M,2M)} \cdot \mathbf{D}^{(1)T} + \sigma_v^2 \cdot \mathbf{I}_M \tag{III.39}$$

where $\tilde{\mathbf{D}}^{(0)} = \mathbf{D}^{(1)}$ (see Equation II.30). The product $\mathbf{D}^{(1)} \cdot \mathbf{D}^{(1)T}$ effectively divides the matrix dimensions by two. Actually, whether we use $\mathbf{D}^{(0)} \cdot \mathbf{D}^{(0)T}$ or $\mathbf{D}^{(1)} \cdot \mathbf{D}^{(1)T}$ it is irrelevant for the derivation of Equation III.39, but we keep the first product for consistency. The fact that $y[m]$ comes from decimating $x[n]$ guarantees that it is WSS, since $R_{yy}[l] = R_{xx}[2l]$.

The cross-correlation term between the high and the low sampling rate signals is given by:

$$\begin{aligned}
\mathbf{R}_{xy}^{(k)*} &= E\left\{\tilde{\mathbf{x}}_{2m+k} \cdot \tilde{\mathbf{y}}_m^{*T}\right\} = E\left\{(\tilde{\mathbf{s}}_{2m+k} + \tilde{\mathbf{u}}_{2m+k}) \cdot \left(\tilde{\mathbf{D}}^{(k)} \cdot \tilde{\mathbf{s}}_{2m+k} + \tilde{\mathbf{v}}_m\right)^{*T}\right\} \\
&= E\left\{\tilde{\mathbf{s}}_{2m+k} \cdot \tilde{\mathbf{s}}_{2m+k}^{*T} \cdot \tilde{\mathbf{D}}^{(k)T}\right\} + E\left\{\tilde{\mathbf{s}}_{2m+k} \cdot \tilde{\mathbf{v}}_m^{*T}\right\} \\
&\quad + E\left\{\tilde{\mathbf{u}}_{2m+k} \cdot \tilde{\mathbf{s}}_{2m+k}^{*T} \cdot \tilde{\mathbf{D}}^{(k)T}\right\} + E\left\{\tilde{\mathbf{u}}_{2m+k} \cdot \tilde{\mathbf{v}}_m^{*T}\right\} \\
&= \tilde{\mathbf{R}}_{ss}^{(N,2M)} \cdot \tilde{\mathbf{D}}^{(k)T}
\end{aligned} \tag{III.40}$$

which implies

$$\mathbf{R}_{xy}^{(k)} = \mathbf{R}_{ss}^{(N,2M)} \cdot \tilde{\mathbf{D}}^{(k)T}. \tag{III.41}$$

We can also express the cross-correlation vectors, in the left-hand side of Equation III.16, in a similar fashion as:

$$\begin{aligned}
\tilde{\mathbf{r}}_{sx} &= E\left\{s[2m+k] \cdot \tilde{\mathbf{x}}_{2m+k}^{*}\right\} = E\left\{s[2m+k] \cdot (\tilde{\mathbf{s}}_{2m+k}^{*} + \tilde{\mathbf{u}}_{2m+k}^{*})\right\} \\
&= E\left\{s[2m+k] \cdot \tilde{\mathbf{s}}_{2m+k}^{*}\right\} + E\left\{s[2m+k] \cdot \tilde{\mathbf{u}}_{2m+k}^{*}\right\} \\
&= \tilde{\mathbf{r}}_{ss}^{(N,1)}
\end{aligned} \tag{III.42}$$

and

$$\begin{aligned}\tilde{\mathbf{r}}_{sy} &= E\left\{s[2m+k]\cdot\tilde{\mathbf{y}}_m^*\right\} = E\left\{s[2m+k]\cdot\left(\tilde{\mathbf{D}}^{(k)}\cdot\tilde{\mathbf{s}}^*_{2m+k}+\tilde{\mathbf{v}}_m\right)\right\}\\ &= \tilde{\mathbf{D}}^{(k)}\cdot E\left\{s[2m+k]\cdot\tilde{\mathbf{s}}^*_{2m+k}\right\}+E\left\{s[2m+k]\cdot\tilde{\mathbf{v}}_m^*\right\}\\ &= \tilde{\mathbf{D}}^{(k)}\cdot\tilde{\mathbf{r}}_{ss}^{(2M,1)}.\end{aligned} \tag{III.43}$$

Substituting Equations III.37 through III.43 into Equation III.16, we obtain:

$$\left[\begin{array}{c|c}\mathbf{R}_{ss}^{(N,N)}+\sigma_u^2\cdot\mathbf{I}_N & \mathbf{R}_{ss}^{(N,2M)}\cdot\tilde{\mathbf{D}}^{(k)T}\\ \\ \tilde{\mathbf{D}}^{(k)}\cdot\left(\mathbf{R}_{ss}^{*(N,2M)}\right)^{*T} & \mathbf{D}^{(1)}\cdot\mathbf{R}_{ss}^{(2M,2M)}\cdot\mathbf{D}^{(1)T}+\sigma_v^2\cdot\mathbf{I}_M\end{array}\right]\cdot\left[\begin{array}{c}\mathbf{h}^{(k)}\\ \\ \mathbf{g}^{(k)}\end{array}\right]=\left[\begin{array}{c}\tilde{\mathbf{r}}_{ss}^{(N,1)}\\ \\ \tilde{\mathbf{D}}^{(k)}\cdot\tilde{\mathbf{r}}_{ss}^{(2M,1)}\end{array}\right]. \tag{III.44}$$

Equation III.44 is the form of the Wiener-Hopf equation for the multirate Wiener filter when $\Theta(z)=\Gamma(z)=1$.

The structure of the correlation matrices $\mathbf{R}_{xx}$, $\mathbf{R}_{yy}$ and $\mathbf{R}_{xy}^{(k)}$ is as follows:

$$\mathbf{R}_{xx}=\begin{bmatrix}R_{ss}[0] & R_{ss}[-1] & \cdots & R_{ss}[-N+1]\\ R_{ss}[1] & R_{ss}[0] & \cdots & R_{ss}[-N+2]\\ \vdots & \vdots & \ddots & \vdots\\ R_{ss}[N-1] & R_{ss}[N-2] & \cdots & R_{ss}[0]\end{bmatrix}+\sigma_u^2\cdot\mathbf{I}_N \tag{III.45}$$

while $\mathbf{R}_{yy}$ is:

$$\mathbf{R}_{yy}=\begin{bmatrix}R_{ss}[0] & R_{ss}[-2] & \cdots & R_{ss}[-2M+2]\\ R_{ss}[2] & R_{ss}[0] & \cdots & R_{ss}[-2M+4]\\ \vdots & \vdots & \ddots & \vdots\\ R_{ss}[2M-2] & R_{ss}[2M-4] & \cdots & R_{ss}[0]\end{bmatrix}+\sigma_v^2\cdot\mathbf{I}_M. \tag{III.46}$$

Finally, $\mathbf{R}_{xy}^{(k)}$ has the form:

$$\mathbf{R}_{xy}^{(k)} = \begin{bmatrix} R_{ss}[0+k] & R_{ss}[-2+k] & \cdots & R_{ss}[-2(M-1)+k] \\ R_{ss}[1+k] & R_{ss}[-1+k] & \cdots & R_{ss}[-2(M-1)+1+k] \\ \vdots & \vdots & \ddots & \vdots \\ R_{ss}[N-1+k] & R_{ss}[N-2+k] & \cdots & R_{ss}[N-1-2(M-1)+k] \end{bmatrix}. \tag{III.47}$$

Similarly, the structure of the cross-correlation vectors $\mathbf{r}_{sx}$ and $\mathbf{r}_{sy}^{(k)}$ is as follows:

$$\mathbf{r}_{sx} = \begin{bmatrix} R_{ss}[0] \\ R_{ss}[1] \\ \vdots \\ R_{ss}[N-1] \end{bmatrix} \tag{III.48}$$

and

$$\mathbf{r}_{sy}^{(k)} = \begin{bmatrix} R_{ss}[0+k] \\ R_{ss}[2+k] \\ \vdots \\ R_{ss}[2M-2+k] \end{bmatrix}. \tag{III.49}$$

2. Filtered Observations

As previously mentioned, consider the original signal $s[n]$ observed in both channels through FIR filters with impulse responses $\theta[n]$ and $\gamma[n]$ with orders P and Q respectively. We also assume that the noise sequences $u[n]$ and $v[m]$ are not necessarily white. With the tools provided so far, we can obtain the correlation matrix of the combined signal observation vector. In the following, $\mathbf{\Theta}$ and $\mathbf{\Gamma}$ denote the convolution matrices of $\theta[n]$ and of $\gamma[n]$ respectively. Computing each term in Equation III.16 separately, we obtain:

$$\begin{aligned}
\mathbf{R}_{xx}^{*} &= E\left\{\tilde{\mathbf{x}}_{2m+k}\cdot\tilde{\mathbf{x}}_{2m+k}^{*T}\right\} = E\left\{\left(\mathbf{\Theta}^{*}\cdot\tilde{\mathbf{s}}_{2m+k}+\tilde{\mathbf{u}}_{2m+k}\right)\cdot\left(\tilde{\mathbf{s}}_{2m+k}^{*T}\cdot\mathbf{\Theta}^{T}+\tilde{\mathbf{u}}_{2m+k}^{*T}\right)\right\} \\
&= \mathbf{\Theta}^{*}\cdot E\left\{\tilde{\mathbf{s}}_{2m+k}\cdot\tilde{\mathbf{s}}_{2m+k}^{*T}\right\}\cdot\mathbf{\Theta}^{T}+\mathbf{\Theta}^{*}\cdot E\left\{\tilde{\mathbf{s}}_{2m+k}\cdot\tilde{\mathbf{u}}_{2m+k}^{*T}\right\} \\
&\quad +E\left\{\tilde{\mathbf{u}}_{2m+k}\cdot\tilde{\mathbf{s}}_{2m+k}^{*T}\right\}\cdot\mathbf{\Theta}^{T}+E\left\{\tilde{\mathbf{u}}_{2m+k}\cdot\tilde{\mathbf{u}}_{2m+k}^{*T}\right\} \\
&= \mathbf{\Theta}^{*}\cdot\tilde{\mathbf{R}}_{ss}^{(N',N')}\cdot\mathbf{\Theta}^{T}+\tilde{\mathbf{R}}_{u}^{(N,N)}
\end{aligned} \tag{III.50}$$

which implies

$$\mathbf{R}_{xx}=\mathbf{\Theta}\cdot\mathbf{R}_{ss}^{(N',N')}\cdot\mathbf{\Theta}^{*T}+\mathbf{R}_{u}^{(N,N)} \tag{III.51}$$

where the term $N' = N - P + 1$ denotes the dimensions that the matrix $\mathbf{R}_{ss}$ should have in order for $\mathbf{R}_{xx}$ to have dimensions $N \times N$. The term $\mathbf{\Theta}^{*}\cdot\tilde{\mathbf{s}}_{2m+k}$ represents reverse filtering on the sequence $s[n]$. On the other hand, the term $\mathbf{\Theta}\cdot\mathbf{R}_{ss}\cdot\mathbf{\Theta}^{*T}$ denotes both forward (zero-phase) and reverse filtering.

Similarly the other diagonal block becomes:

$$\begin{aligned}
\mathbf{R}_{yy}^{*} &= E\left\{\tilde{\mathbf{y}}_{m}\cdot\tilde{\mathbf{y}}_{m}^{*T}\right\} = E\left\{\left(\tilde{\mathbf{D}}^{(0)}\cdot\mathbf{\Gamma}^{*}\cdot\tilde{\mathbf{s}}_{2m+k}+\tilde{\mathbf{v}}_{m}\right)\cdot\left(\tilde{\mathbf{s}}_{2m+k}^{*T}\cdot\mathbf{\Gamma}^{T}\cdot\tilde{\mathbf{D}}^{(0)*T}+\tilde{\mathbf{v}}_{m}^{*T}\right)\right\} \\
&= E\left\{\mathbf{D}^{(1)}\cdot\mathbf{\Gamma}^{*}\cdot\tilde{\mathbf{s}}_{2m+k}\cdot\tilde{\mathbf{s}}_{2m+k}^{*T}\cdot\mathbf{\Gamma}^{T}\cdot\mathbf{D}^{(1)T}\right\}+E\left\{\mathbf{D}^{(1)}\cdot\mathbf{\Gamma}^{*}\cdot\tilde{\mathbf{s}}_{2m+k}\cdot\tilde{\mathbf{v}}_{m}^{*T}\right\} \\
&\quad +E\left\{\tilde{\mathbf{v}}_{m}\cdot\tilde{\mathbf{s}}_{2m+k}^{*T}\cdot\mathbf{\Gamma}^{T}\cdot\mathbf{D}^{(1)T}\right\}+E\left\{\tilde{\mathbf{v}}_{m}\cdot\tilde{\mathbf{v}}_{m}^{*T}\right\} \\
&= \mathbf{D}^{(1)}\cdot\mathbf{\Gamma}^{*}\cdot\tilde{\mathbf{R}}_{ss}^{(M',M')}\cdot\mathbf{\Gamma}^{T}\cdot\mathbf{D}^{(1)T}+\tilde{\mathbf{R}}_{v}^{(M,M)}
\end{aligned} \tag{III.52}$$

which is written

$$\mathbf{R}_{yy}=\mathbf{D}^{(1)}\cdot\mathbf{\Gamma}\cdot\mathbf{R}_{ss}^{(M',M')}\cdot\mathbf{\Gamma}^{*T}\cdot\mathbf{D}^{(1)T}+\mathbf{R}_{v}^{(M,M)} \tag{III.53}$$

where the term $M' = 2M - Q + 1$ denotes the dimensions that the matrix $\mathbf{R}_{ss}$ should have in order for $\mathbf{R}_{yy}$ to have dimensions $M \times M$.

The cross-term can be computed as:

$$\begin{aligned}
\mathbf{R}_{xy}^{(k)*} &= E\left\{\tilde{\mathbf{x}}_{2m+k} \cdot \tilde{\mathbf{y}}_m^{*T}\right\} \\
&= E\left\{\left(\boldsymbol{\Theta}^* \cdot \tilde{\mathbf{s}}_{2m+k} + \tilde{\mathbf{u}}_{2m+k}\right) \cdot \left(\tilde{\mathbf{D}}^{(k)T} \cdot \boldsymbol{\Gamma}^* \cdot \tilde{\mathbf{s}}_{2m+k} + \tilde{\mathbf{v}}_m^{*T}\right)^{*T}\right\} \\
&= \boldsymbol{\Theta}^* \cdot E\left\{\tilde{\mathbf{s}}_{2m+k} \cdot \tilde{\mathbf{s}}_{2m+k}^{*T}\right\} \cdot \boldsymbol{\Gamma}^T \cdot \tilde{\mathbf{D}}^{(k)T} + \boldsymbol{\Theta}^* \cdot E\left\{\tilde{\mathbf{s}}_{2m+k} \cdot \tilde{\mathbf{v}}_m^{*T}\right\} \\
&\quad + E\left\{\tilde{\mathbf{u}}_{2m+k} \cdot \tilde{\mathbf{s}}_{2m+k}^{*T}\right\} \cdot \boldsymbol{\Gamma}^T \cdot \tilde{\mathbf{D}}^{(k)T} + E\left\{\tilde{\mathbf{u}}_{2m+k} \cdot \tilde{\mathbf{v}}_m^{*T}\right\} \\
&= \boldsymbol{\Theta}^* \cdot \tilde{\mathbf{R}}_{ss}^{(N',M')} \cdot \boldsymbol{\Gamma}^T \cdot \tilde{\mathbf{D}}^{(k)T}
\end{aligned} \tag{III.54}$$

which yields

$$\mathbf{R}_{xy}^{(k)} = \boldsymbol{\Theta} \cdot \mathbf{R}_{ss}^{(N',M')} \cdot \boldsymbol{\Gamma}^{*T} \cdot \tilde{\mathbf{D}}^{(k)T}. \tag{III.55}$$

Finally,

$$\mathbf{R}_{xy}^{(k)*T} = \tilde{\mathbf{D}}^{(k)} \cdot \boldsymbol{\Gamma} \cdot \left(\mathbf{R}_{ss}^{*(N',M')}\right)^{*T} \cdot \boldsymbol{\Theta}^{*T}. \tag{III.56}$$

By combining Equations III.51 through III.56, the left-hand side in Equation III.16 becomes:

$$\mathbf{R}^{(k)} = \left[\begin{array}{c|c}
\boldsymbol{\Theta} \cdot \mathbf{R}_{ss}^{(N',N')} \cdot \boldsymbol{\Theta}^{*T} + \mathbf{R}_u^{(N,N)} & \boldsymbol{\Theta} \cdot \mathbf{R}_{ss}^{(N',M')} \cdot \boldsymbol{\Gamma}^{*T} \cdot \tilde{\mathbf{D}}^{(k)T} \\
 & \\
\tilde{\mathbf{D}}^{(k)} \cdot \boldsymbol{\Gamma} \cdot \left(\mathbf{R}_{ss}^{*(N',M')}\right)^{*T} \cdot \boldsymbol{\Theta}^{*T} & \mathbf{D}^{(1)} \cdot \boldsymbol{\Gamma} \cdot \mathbf{R}_{ss}^{(M',M')} \cdot \boldsymbol{\Gamma}^{*T} \cdot \mathbf{D}^{(1)T} + \mathbf{R}_v^{(M,M)}
\end{array}\right]. \tag{III.57}$$

The cross-correlation vectors in the right-hand side of Equation III.16 become:

$$\begin{aligned} \tilde{\mathbf{r}}_{sx} &= E\left\{s[n]\cdot\tilde{\mathbf{x}}_n^*\right\} = E\left\{s[n]\cdot\left(\boldsymbol{\Theta}^*\cdot\tilde{\mathbf{s}}_n+\tilde{\mathbf{u}}_n\right)^*\right\} \\ &= \boldsymbol{\Theta}\cdot E\left\{s[n]\cdot\tilde{\mathbf{s}}_n^*\right\} + E\left\{s[n]\cdot\tilde{\mathbf{u}}_n^*\right\} \end{aligned} \tag{III.58}$$

which implies

$$\tilde{\mathbf{r}}_{sx} = \boldsymbol{\Theta}\cdot\tilde{\mathbf{r}}_{ss}^{(N',1)}. \tag{III.59}$$

Similarly

$$\begin{aligned} \tilde{\mathbf{r}}_{sy}^{(k)} &= E\left\{s[n]\cdot\tilde{\mathbf{y}}_m^*\right\} = E\left\{s[n]\cdot\left(\tilde{\mathbf{D}}^{(k)}\cdot\boldsymbol{\Gamma}^*\cdot\tilde{\mathbf{s}}_n+\tilde{\mathbf{v}}_m\right)^*\right\} \\ &= \tilde{\mathbf{D}}^{(k)}\cdot\boldsymbol{\Gamma}\cdot E\left\{s[n]\cdot\tilde{\mathbf{s}}_n^*\right\} + E\left\{s[n]\cdot\tilde{\mathbf{v}}_m^*\right\} \end{aligned} \tag{III.60}$$

which becomes

$$\tilde{\mathbf{r}}_{sy}^{(k)} = \tilde{\mathbf{D}}^{(k)}\cdot\boldsymbol{\Gamma}\cdot\tilde{\mathbf{r}}_{ss}^{(M',1)}. \tag{III.61}$$

We substitute Equations III.57, III.59 and III.61, in Equation III.16 and obtain:

$$\mathbf{R}^{(k)}\cdot\begin{bmatrix}\mathbf{h}^{(k)} \\ \\ \mathbf{g}^{(k)}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\Theta}\cdot\tilde{\mathbf{r}}_{ss}^{(N',1)} \\ \\ \tilde{\mathbf{D}}^{(k)}\cdot\boldsymbol{\Gamma}\cdot\tilde{\mathbf{r}}_{ss}^{(M',1)}\end{bmatrix} \tag{III.62}$$

where $\mathbf{R}^{(k)}$ is given by Equation III.57. The significance of Equation III.62 is that it provides a mean of computing the multirate Wiener filter coefficients $\mathbf{h}^{(k)}$ and $\mathbf{g}^{(k)}$ for $k = 0, 1$, directly from the signal statistics ($\mathbf{R}_{ss}$) and the filter impulse responses ($\boldsymbol{\Theta}$ and $\boldsymbol{\Gamma}$).

A closer look at Equation III.62, reveals that part of it incorporates the single rate Wiener filter. Thus, we can produce a more general form for it, expressed from

Equation III.63 that represents any signal that is subject to linear transformation (by $\theta[n]$) prior to the computation of the Wiener filter.

$$(\mathbf{\Theta} \cdot \mathbf{R}_{ss} \cdot \mathbf{\Theta}^{*T} + \mathbf{R}_u) \cdot \mathbf{h} = \mathbf{\Theta} \cdot \tilde{\mathbf{r}}_{ss} \tag{III.63}$$

D. OTHER RELATED MULTIRATE PROBLEMS

In the previous Chapter we proved that the estimate of the Multirate Wiener filter is optimal for the general problem illustrated in Figure 6. The next step will be to show how the multirate Wiener filter of Figure 8 can be modified and used in slightly different scenarios. A close examination of Figure 8 reveals two possible modifications.

1. Perform decimation prior to filtering. The modified block diagram appears in Figure 9.
2. Extend the result to decimation by a factor of K where K is an integer greater than or equal to 2.

1. Decimation Prior to Filtering

When the signal is decimated prior to filtering, also shown in Figure 9, the resulting observation vector $\tilde{\mathbf{y}}_m$ is computed as:

$$\tilde{\mathbf{y}}_m = \mathbf{\Gamma}^* \cdot \tilde{\mathbf{D}}^{(0)} \cdot \tilde{\mathbf{s}}_{2m+k}. \tag{III.64}$$

The multirate Wiener-Hopf equation then has the explicit form:

$$\mathbf{R}^{(k)} \cdot \begin{bmatrix} \mathbf{h}^{(k)} \\ \\ \mathbf{g}^{(k)} \end{bmatrix} = \begin{bmatrix} \mathbf{\Theta} \cdot \tilde{\mathbf{r}}_{ss}^{(N',1)} \\ \\ \mathbf{\Gamma} \cdot \tilde{\mathbf{D}}^{(k)} \cdot \tilde{\mathbf{r}}_{ss}^{(M',1)} \end{bmatrix} \tag{III.65}$$

where

$$\mathbf{R}^{(k)} = \left[\begin{array}{c|c} \boldsymbol{\Theta} \cdot \mathbf{R}_{ss}^{(N',N')} \cdot \boldsymbol{\Theta}^{*T} + \mathbf{R}_{u}^{(N,N)} & \boldsymbol{\Theta} \cdot \mathbf{R}_{ss}^{(N',M')} \cdot \tilde{\mathbf{D}}^{(k)T} \cdot \boldsymbol{\Gamma}^{*T} \\ & \\ \boldsymbol{\Gamma} \cdot \tilde{\mathbf{D}}^{(k)} \cdot \mathbf{R}_{ss}^{*(M',N')} \cdot \boldsymbol{\Theta}^{*T} & \mathbf{D}^{(1)} \cdot \boldsymbol{\Gamma} \cdot \mathbf{R}_{ss}^{(M',M')} \cdot \boldsymbol{\Gamma}^{*T} \cdot \mathbf{D}^{(1)T} + \mathbf{R}_{v}^{(M,M)} \end{array} \right] \tag{III.66}$$

and the dimensions are $N' = N - P + 1$ and $M' = 2M - 2Q + 2$.

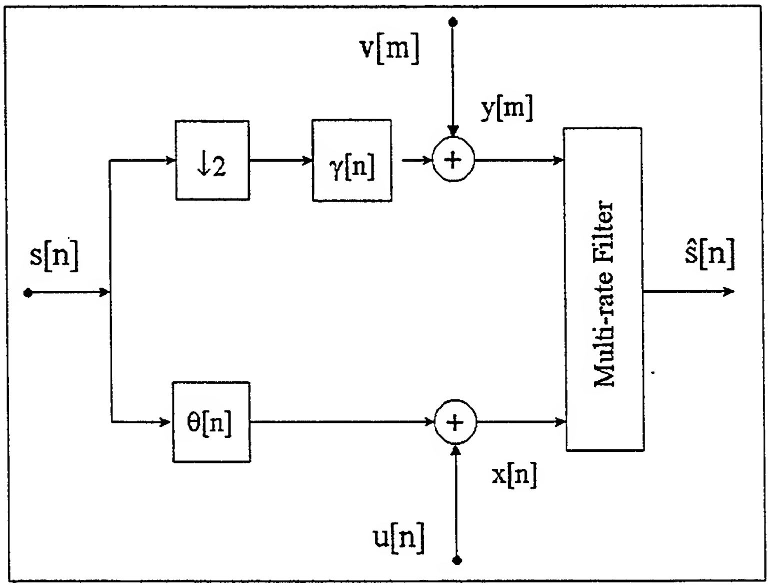

Figure 9. Block Diagram with Filtering Prior to Decimation

2. Decimation by a Factor of K

Let us return to the system of Figure 8 and suppose that the decimation is performed by a factor of K, where $K \geq 2$. We can generalize the decimation matrix as $\mathbf{D}_K^{(k)}$, where the subscript K denotes the decimation factor and the superscript $k = 0, 1, 2, \ldots, K-1$ denotes the time delay that the matrix introduces. All matrices $\mathbf{D}_K^{(k)}$ have dimensions $M \times K \cdot M$ and can be computed as follows:

$$\mathbf{D}_K^{(k)} = \begin{bmatrix} \underbrace{0,\cdots,\overbrace{1,0,\cdots,0}^{k+1}}_{K} & 0 & \cdots & 0 \\ \ddots & \ddots & \ddots & \ddots \\ 0 & \cdots & 0 & \underbrace{0,\cdots,\overbrace{1,0,\cdots,0}^{k+1}}_{K} \end{bmatrix} . \tag{III.67}$$

All the identities that were valid for $K = 2$, can be generalized to any value $K > 2$ as:

$$\mathbf{U}_K^{(k)} = \mathbf{D}_K^{(k)T} \tag{III.68}$$

and

$$\tilde{\mathbf{D}}_K^{(k)} = \mathbf{D}_K^{(K-1-k)} \tag{III.69}$$

where $k = 0, 1, 2, \ldots, K-1$. The multirate Wiener-Hopf equation becomes:

$$\mathbf{R}^{(k)} \cdot \begin{bmatrix} \mathbf{h}^{(k)} \\ \\ \mathbf{g}^{(k)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Theta} \cdot \tilde{\mathbf{r}}_{ss}^{(N',1)} \\ \\ \boldsymbol{\Gamma} \cdot \tilde{\mathbf{D}}^{(k)} \cdot \tilde{\mathbf{r}}_{ss}^{(M',1)} \end{bmatrix} \tag{III.70}$$

where

$$\mathbf{R}^{(k)} = \begin{bmatrix} \boldsymbol{\Theta} \cdot \mathbf{R}_{ss}^{(N',N')} \cdot \boldsymbol{\Theta}^{*T} + \mathbf{R}_u^{(N,N)} & | & \boldsymbol{\Theta} \cdot \mathbf{R}_{ss}^{(N',M')} \cdot \boldsymbol{\Gamma}^{*T} \cdot \tilde{\mathbf{D}}_K^{(k)T} \\ & | & \\ \tilde{\mathbf{D}}_K^{(k)} \cdot \boldsymbol{\Gamma} \cdot \mathbf{R}_{ss}^{*(M',N')} \cdot \boldsymbol{\Theta}^{*T} & | & \mathbf{D}_K^{(K-1)} \cdot \boldsymbol{\Gamma} \cdot \mathbf{R}_{ss}^{(M',M')} \cdot \boldsymbol{\Gamma}^{*T} \cdot \mathbf{D}_K^{(K-1)T} + \mathbf{R}_v^{(M,M)} \end{bmatrix} \tag{III.71}$$

and the dimensions are $N' = N - P + 1$ and $M' = KM - Q + 1$.

The structure of the matrix equation depends on two factors. The correlation matrix has a 2×2 block structure that results from the presence of two observation sequences. In a more general situation of multiple observation sequences, the number of blocks would be equal to the number of those sequences. For instance, three observations would lead to a 3×3 structure of the correlation matrix and three distinct filtering functions, one for each observation sequence. On the other hand, the number of times that the filter has to be computed, depends on the number of samples between the observation on the high sampling rate and the observation on the low sampling rate. As an example, if we obtain three observations at decimation factors $D_1 = 1$ (no decimation), $D_2 = 2$ and $D_3 = 4$ of the original sampling rate, the multirate Wiener filter would involve three distinct filtering functions (say $\mathbf{h}_1^{(k)}$, $\mathbf{h}_2^{(k)}$ and $\mathbf{h}_3^{(k)}$), and would have to be computed for $k = 0, 1, 2, 3$ in order to cover the new observations from all signals.

E. EXPERIMENTAL RESULTS

The experiments were based on simulations created with the use of MATLAB. For the needs of the filter implementation, a function was created with the name MultiWiener.m . The source file is included in the Appendix, while the source code is included in an enclosed diskette. In all cases, the model that was used was a second order AR (auto-regressive) model with transfer function $1/A(z)$ and generating noise variance $\sigma_w^2 = 1$ as shown in Figure 10. The experiments were divided into two parts:

1. The use of the function only in order to check its theoretical performance. In this case, the results included the filter coefficients and the computed minimum MSE, using different combinations of additive white noise. The performance was compared to that of the single rate Wiener filter.

2. A set of two simulations where a signal was created, distorted and filtered according to the model. The first simulation was conducted without pre-filtering. For the second simulation, the pre-filtering transfer functions were obtained from the use of the *Remez Algorithm.* Both transfer functions described low-pass filters of different orders. The filter performance was compared to that of the single rate Wiener filter.

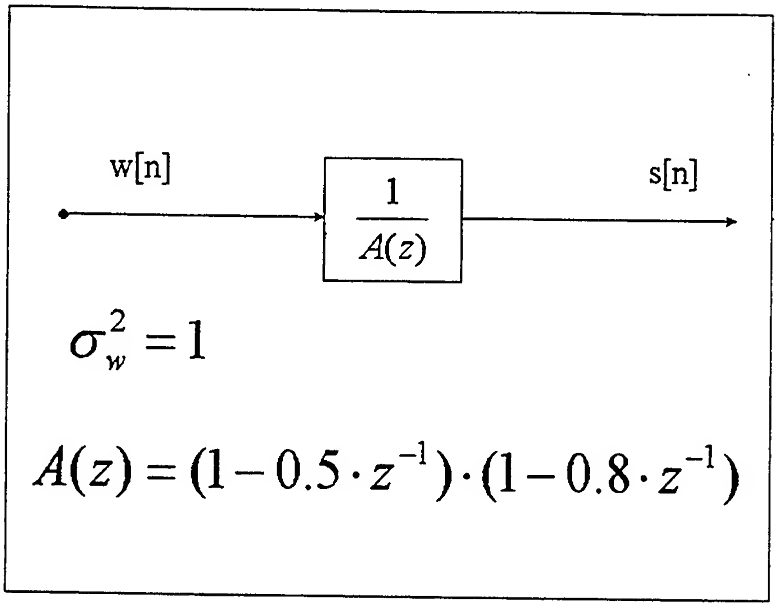
w[n]
1
A(z)
s[n]
$\sigma_w^2 = 1$
$A(z) = (1 - 0.5 \cdot z^{-1}) \cdot (1 - 0.8 \cdot z^{-1})$

Figure 10. AR Model for Signal $s[n]$

1. Theoretical Performance

For the first set of results without pre-filtering ($\Theta(z) = 1$ and $\Gamma(z) = 1$), we considered several cases of different additive white noise variances σ_u^2 and σ_v^2. The filter orders for $\mathbf{h}^{(k)}$ and $\mathbf{g}^{(k)}$ remained the same ($N = 6$ and $M = 4$) for all cases. In addition, instead of presenting the filter vectors $\mathbf{h}^{(k)}$ and $\mathbf{g}^{(k)}$, the following tables present their 2-norm ($||\mathbf{h}||_2$) as a metric for the weight of the filter coefficients. The results from this set are shown in Table III.

A close look at Table III, provides some interesting observations.

1. In the first case where both $\sigma_u^2 = 1$ and $\sigma_v^2 = 1$, $\mathbf{h}^{(1)}$ has much larger weight than $\mathbf{g}^{(1)}$, since only the measurement from $x[n]$ is present. For $k = 0$ on the other hand, both filters have similar weights, since there are available measurements from both signals.
2. In the second and third cases, where $\sigma_u^2 = 0$ and $\sigma_v^2 = 0$ respectively, the weight of the filter coefficients is transfered completely to the respective filter ($\mathbf{h}^{(k)}$ or $\mathbf{g}^{(k)}$), since the additive noise is zero.

Index k	σ_ε^2	$\lVert \mathbf{h}^{(k)} \rVert_2$	$\lVert \mathbf{g}^{(k)} \rVert_2$
$\sigma_u^2 = 1,\ \sigma_v^2 = 1$			
0	0.3959	0.4243	0.3965
1	0.6116	0.6369	0.1768
Wiener	0.6572	0.7079	–
$\sigma_u^2 = 0,\ \sigma_v^2 = 1$			
0	$2.3093 \cdot 10^{-14}$	1.0	$2.2204 \cdot 10^{-16}$
1	$5.8637 \cdot 10^{-14}$	1.0	$6.6717 \cdot 10^{-16}$
Wiener	$2.3093 \cdot 10^{-14}$	1.0	–
$\sigma_u^2 = 1,\ \sigma_v^2 = 0$			
0	$8.8818 \cdot 10^{-15}$	$9.0180 \cdot 10^{-16}$	1.0
1	0.5107	0.5136	0.5731
Wiener	0.6572	0.7078	–
$\sigma_u^2 = 1,\ \sigma_v^2 = 1000$			
0	0.6567	0.7074	$6.5884 \cdot 10^{-4}$
1	0.6571	0.7078	$2.5763 \cdot 10^{-4}$
Wiener	0.6572	0.7078	–
$\sigma_u^2 = 1000,\ \sigma_v^2 = 1$			
0	0.7808	$9.3373 \cdot 10^{-4}$	0.7980
1	1.9546	0.0021	0.8211
Wiener	8.3708	0.0161	–

Table III. Theoretical Multirate Filter Performance

3. Finally in the fourth and fifth cases, where $\sigma_v^2 = 1000$ and $\sigma_u^2 = 1000$ respectively, the weight of the filter coefficients is also transfered by the most part according to what data has the smallest additive noise variance; that is why for both $k = 0$ and $k = 1$ the MSE is about the same when $\sigma_v^2 = 1000$. The value of the second observation is evident in the last case, where the single rate Wiener filter has no effect and the resulting MSE is almost as much as the signal variance ($R_{ss}[0]$).

2. Simulations

The main purpose of the simulations was to confirm the theoretical predictions, by comparing the performance of both filters on simulated data. In both cases, the orders of the multirate Wiener filter were chosen as $N = 12$ and $M = 8$, while the order for the single rate Wiener filter was $N + M = 12 + 8 = 20$. Also, for both simulations, the additive white noise variances were $\sigma_u^2 = \sigma_v^2 = 1$. In both simulations, the data generation was repeated 50 times. The following figures show the time averaged squared error for both the multirate and the single rate Wiener filter in each of the 50 experiments.

The performance comparison for the first simulation, which was conducted without pre-filtering is shown in Figure 11. The theoretical minimum MSE from the multirate filter was $\sigma_\varepsilon^2 = 0.3959$ for $k = 0$ and $\sigma_\varepsilon^2 = 0.6116$ for $k = 1$, yielding an average of 0.5038. On the other hand, the theoretical minimum MSE from the single rate Wiener filter was $\sigma_0^2 = 0.6572$. The average computed from the 50 time averaged square errors shown in Figure 11, matches the σ_ε^2 values quite accurately.

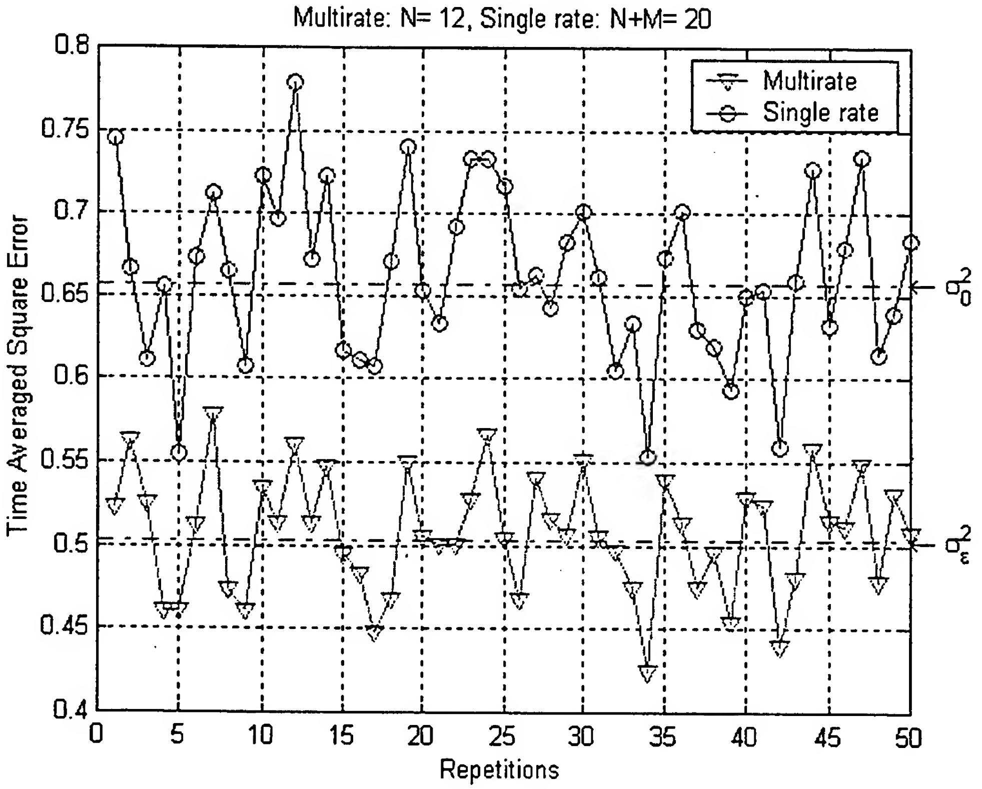

Figure 11. Performance Comparison without Pre-filtering

The second simulation was conducted with pre-filtering, using the Remez Algorithm in order to model the transfer functions. For the design of $\Theta(\omega)$ the order was $P = 6$ and cut-off frequency has been chosen as 0.4π. For the design of $\Gamma(\omega)$ the order was $Q = 12$ and cut-off frequency 0.3π. Both selected frequency responses are presented in Figure 12. The performance comparison between the two filters is shown in Figure 13. The theoretical minimum MSE from the multirate filter was $\sigma_\varepsilon^2 = 0.3801$ for $k = 0$ and $\sigma_\varepsilon^2 = 0.4093$ for $k = 1$. The average of the two previous values is $\sigma_\varepsilon^2 = 0.3947$. On the other hand, the theoretical minimum MSE from the single rate Wiener filter was $\sigma_\varepsilon^2 = 0.4665$. The theoretical performance and the average from Figure 13 are very close.

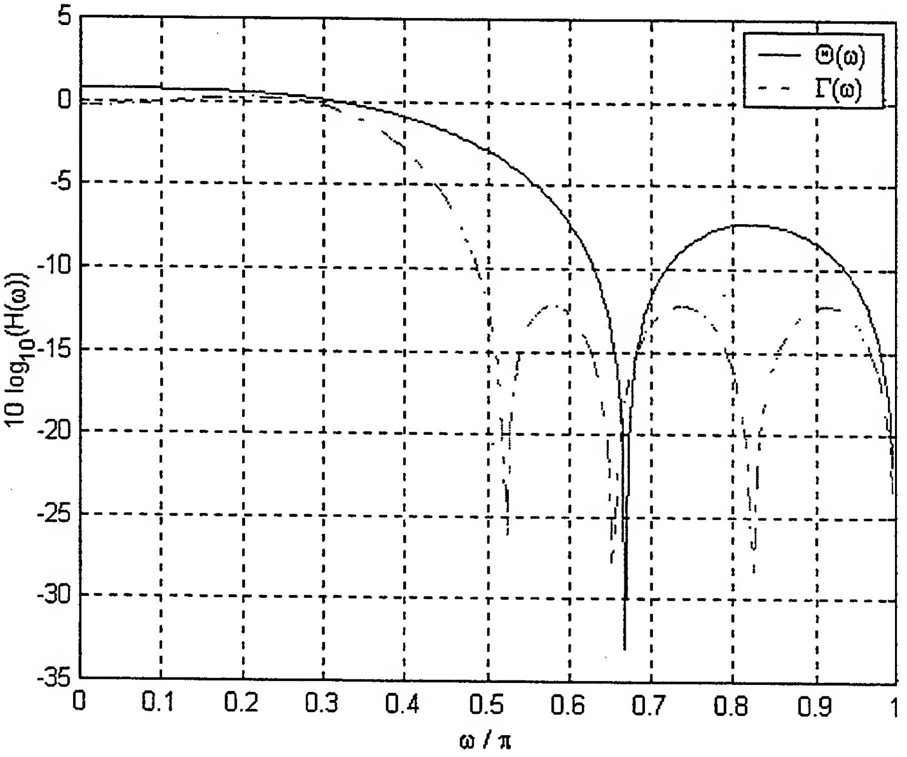

Figure 12. Magnitude Frequency Response for $\Theta(\omega)$ and $\Gamma(\omega)$

3. Some Notes on the Simulations

The results from Figures 11 and 13 confirm the theoretical results, that is *the Multirate Wiener filter performance is in all cases superior to that of the single rate Wiener filter.* However, there are some points that should be pointed out about

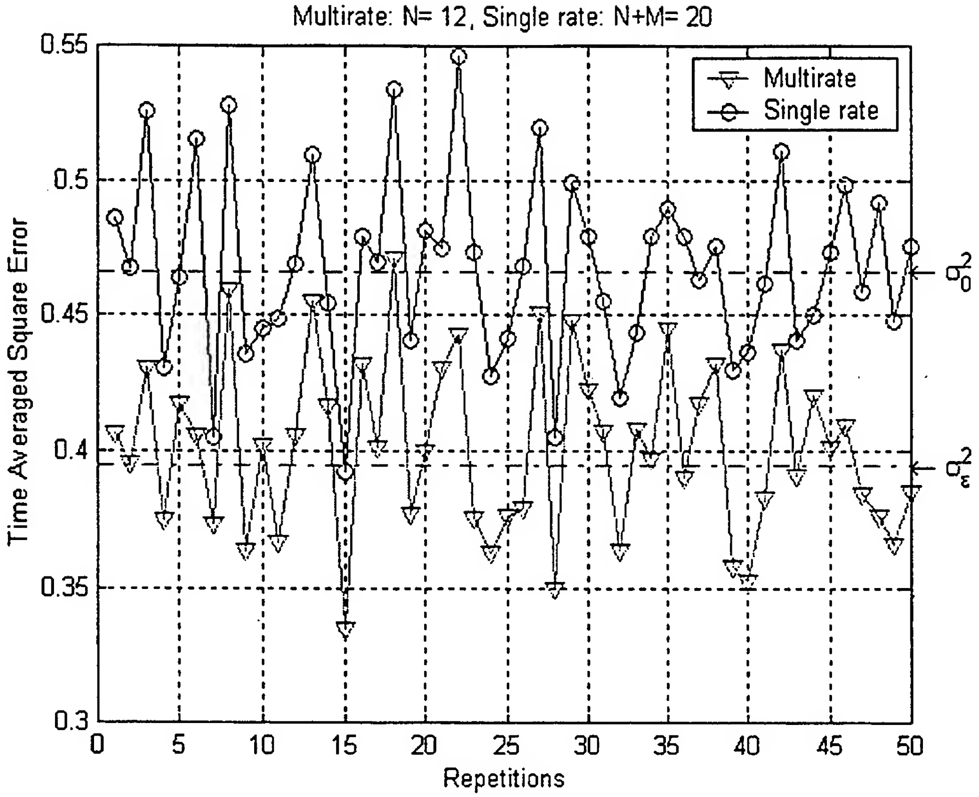

Figure 13. Performance Comparison with Pre-filtering

the way the simulations were implemented. Operations like $\mathbf{R}_{xx} = \Theta \cdot \mathbf{R}_{ss} \cdot \Theta^{*T}$ have the characteristic that they perform both ways, causal and anti-causal, allowing zero shift of the correlation function after filtering. On the other hand, operations like $\mathbf{r}_{sx} = \Theta \cdot \mathbf{r}_{ss}$ cause a shift that turns out to be half the order of the filter (e.g. $P/2$). That shift was taken into account, both during the filter computation as well as during the performing of the simulation. The result was that both filtered versions of $s[n]$, $s_\theta[n]$ and $s_\gamma[n]$, were delayed by $P/2$ and $Q/2$ respectively.

IV. LINEAR PREDICTION FOR MULTIRATE SIGNALS

In many applications, *linear prediction* is used as special type of optimal filtering. Although it is indeed an optimal filtering problem, linear prediction and the sister topic of *autoregressive* (AR) *modeling* are a very rich area with numerous specific applications relating to signal processing. These include coding, spectral analysis, system modeling and others. Basically in linear prediction, the past values of a random process are observed, and it is desired to estimate the current value (which is assumed unknown). Suppose that we form a vector out of the even and odd sample of a signal $s[n]$ called a *signal vector* as:

$$\mathbf{x}[n] = \begin{bmatrix} x[2n] \\ x[2n-1] \end{bmatrix}. \tag{IV.1}$$

The purpose of the analysis of this Chapter is to perform linear prediction on this kind of signal. Since this signal consists of two channels, the techniques used are *multichannel* techniques. We characterize a signal as multichannel, when possibly several correlated signals are received on different channels and then processed and analyzed together. A common example of a multichannel 2-D signal is a color image. The signal is two-dimensional but consists of three registered components representing red, green and blue intensities, or other combinations of values such as those used in various video standards. Although what was previously stated suggests otherwise, the motivation for applying multichannel linear prediction came from the Polyphase Representation.

A. THE POLYPHASE REPRESENTATION

The polyphase representation of a filter, is the foundation of a number of techniques in multirate signal processing. It permits great simplification of theoretical results and also leads to computationally efficient implementations of decimation/interpolation filters, as well as filter banks, both single rate and multirate [Ref. 6]. To explain the basic idea, consider a filter with transfer function $H(z) = \sum_{n=-\infty}^{+\infty} h[n] \cdot z^{-n}$. By separating the even-numbered from the odd-numbered coefficients of $h[n]$, we can write:

$$H(z) = \sum_{n=-\infty}^{+\infty} h[2n] \cdot z^{-2n} + z^{-1} \cdot \sum_{n=-\infty}^{+\infty} h[2n+1] \cdot z^{-2n}. \qquad \text{(IV.2)}$$

Then by defining:

$$H_0(z) = \sum_{n=-\infty}^{+\infty} h[2n] \cdot z^{-n} \;,\; H_1(z) = \sum_{n=-\infty}^{+\infty} h[2n+1] \cdot z^{-n} \qquad \text{(IV.3)}$$

as the filters related to the even and odd samples of the impulse response respectively, we can write $H(z)$ as:

$$H(z) = H_0(z^2) + z^{-1} \cdot H_1(z^2). \qquad \text{(IV.4)}$$

The last representation holds whether $H(z)$ is FIR or IIR; causal or non-causal. It also provides an alternative representation of the filtering operation. Suppose that a *deterministic* signal $x[n]$ is filtered by $H(z)$, and $y[n]$ is the output sequence. If we are interested in the decimated sequence $y[2m]$ only, the polyphase representation becomes particularly efficient, as shown in Figure 14. Similarly, we can delay the input signal by one and use the same technique to obtain $y[2m-1]$ as shown in Figure 15.

The *Noble Identities*, described in Chapter II, are at the basis of the results in Figures 14 and 15. The results that are shown in Figures 16 and 17 show that we can obtain the even or the odd samples of $y[n]$ as a linear combination of filtered even and odd samples of $x[n]$. If we call $X_0(z)$ the $\mathcal{Z}$-transform of $x[2m]$ and $X_1(z)$ the $\mathcal{Z}$-transform of $x[2m-1]$, and $Y_0(z)$ and $Y_1(z)$ the $\mathcal{Z}$-transforms of $y[2m]$ and $y[2m-1]$, we can form a vector from the two signals and represent the operations of both Figures 16 and 17 in matrix form as:

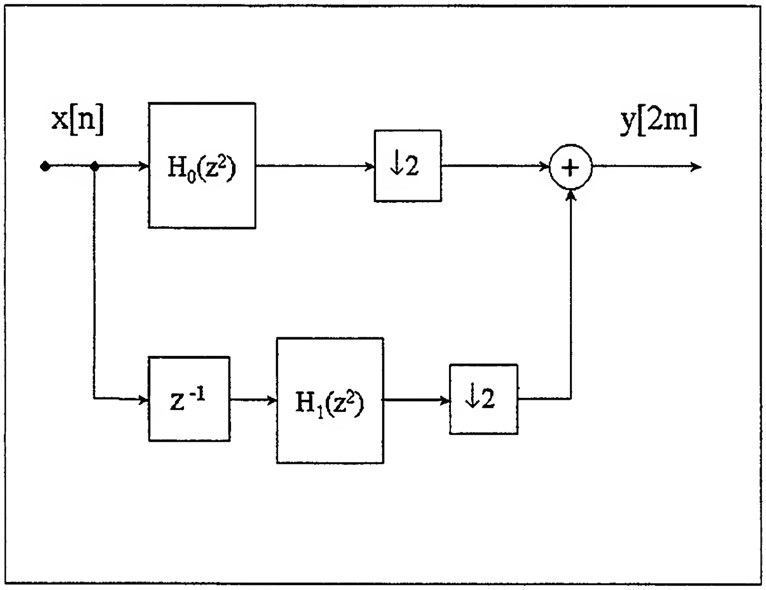

Figure 14. Even Samples $y[2m]$ via the Polyphase Representation

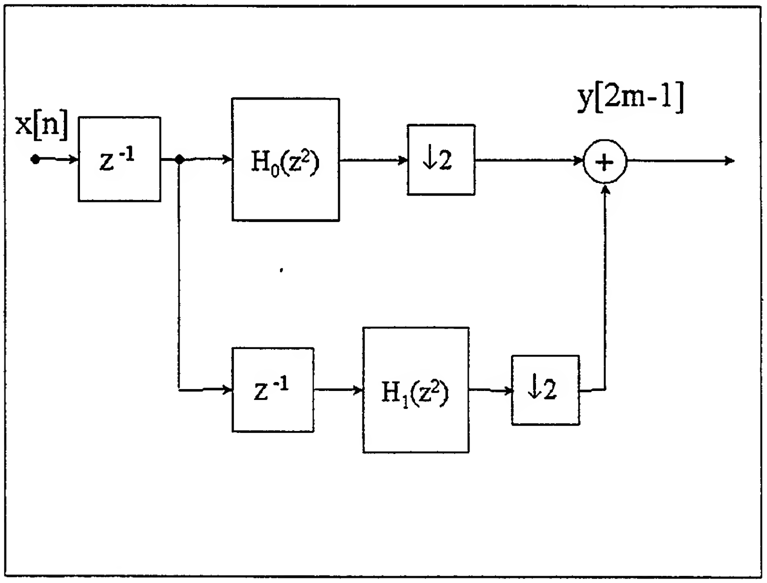

Figure 15. Odd Samples $y[2m-1]$ via the Polyphase Representation

$$\begin{bmatrix} Y_0(z) \\ \\ Y_1(z) \end{bmatrix} = \left[\begin{array}{c|c} H_0(z) & H_1(z) \\ & \\ z^{-1} \cdot H_1(z) & H_0(z) \end{array}\right] \cdot \begin{bmatrix} X_0(z) \\ \\ X_1(z) \end{bmatrix} . \tag{IV.5}$$

The matrix in Equation IV.5 is called the *polyphase matrix.* The importance of this decomposition comes from the fact that the transfer function $H(z)$ is implemented at lower sampling rate. The structure of the polyphase matrix suggests two interesting problems:

1. Devise a method to make the polyphase matrix diagonal, and decompose the even and odd components of $y[n]$ into uncorrelated sequences.
2. Obtain $x_0[n]$ and $x_1[n]$ from $y_0[n]$ and $y_1[n]$. In case the signal is stochastic, this decomposition would yield two innovations, from which the original signal can be reconstructed.

B. MULTICHANNEL RANDOM PROCESSES

A multichannel random process with M channels ([Ref. 11], [Ref. 12], [Ref. 13]) is defined as a vector of M random processes:

$$\mathbf{x}[n] = \begin{bmatrix} x_1[n] \\ x_2[n] \\ \vdots \\ x_M[n] \end{bmatrix} . \tag{IV.6}$$

If all of these signals are jointly stationary, the multichannel autocorrelation function is a sequence of $M \times M$ matrices, defined as:

$$\mathbf{r}_{\mathbf{x}}[l] \overset{def.}{=} E\left\{\mathbf{x}[n] \cdot \mathbf{x}^{*T}[n-l]\right\} . \tag{IV.7}$$

For example, in the case $M = 2$, the ACF is given by

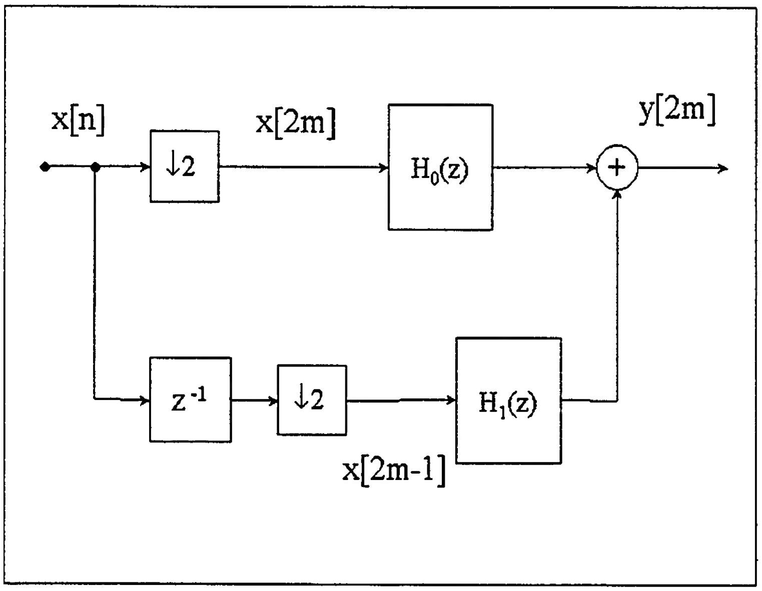

Figure 16. The Noble Identities for $y[2m]$

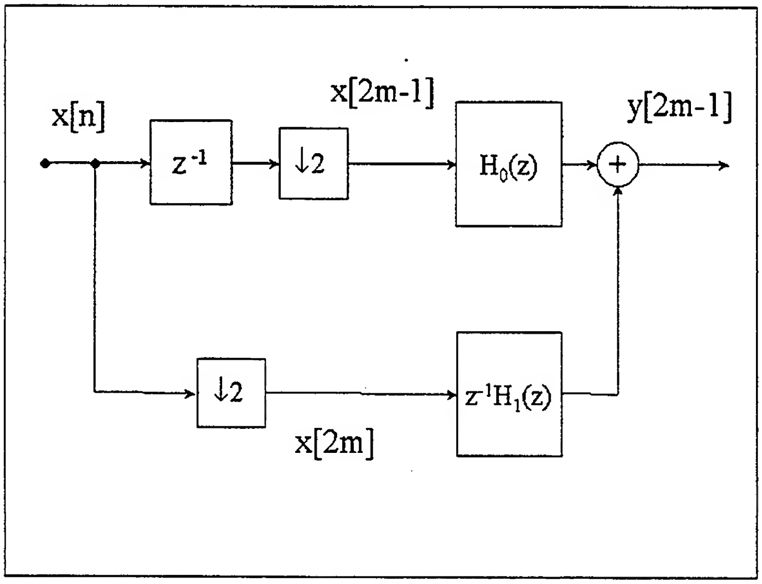

Figure 17. The Noble Identities for $y[2m-1]$

$$\mathbf{r}_{\mathbf{x}}[l] = \begin{bmatrix} R_{x_1,x_1}[l] & R_{x_1,x_2}[l] \\ R_{x_2,x_1}[l] & R_{x_2,x_2}[l] \end{bmatrix}. \qquad \text{(IV.8)}$$

1. Multichannel Linear Prediction

We can extend the concept of linear prediction to the multichannel case ([Ref. 14], [Ref. 13]) and compute the estimate $\hat{\mathbf{x}}[n]$ of a multichannel signal as a linear combination of P past values of the signal. The form of the estimate is:

$$\hat{\mathbf{x}}[n] = -\mathbf{A}_1^{*T} \cdot \mathbf{x}[n-1] - \mathbf{A}_2^{*T} \cdot \mathbf{x}[n-2] - \ldots - \mathbf{A}_P^{*T} \cdot \mathbf{x}[n-P] \qquad \text{(IV.9)}$$

where the coefficients are $M \times M$ matrices, defined as negated and Hermitian transposes for later convenience. The resulting prediction error $\epsilon[n]$, is defined as:

$$\epsilon[n] = \mathbf{x}[n] - \hat{\mathbf{x}}[n] = \sum_{i=0}^{P} \mathbf{A}_i^{*T} \cdot \mathbf{x}[n-i]. \qquad \text{(IV.10)}$$

For the estimate to be optimum in the mean square sense, the resulting mean squared error (MSE) defined as $E\left\{\epsilon[n] \cdot \epsilon^{*T}[n]\right\}$ has to be minimized. The orthogonality principle in the multichannel case requires:

$$E\left\{\mathbf{x}[n-l] \cdot \epsilon^{*T}[n]\right\} = \begin{cases} \mathbf{\Sigma}_{\epsilon}, & l = 0 \\ \mathbf{0}, & l = 1, 2, \ldots, P. \end{cases} \qquad \text{(IV.11)}$$

The matrix $\mathbf{\Sigma}_{\varepsilon}$ is called the *prediction error covariance matrix* and is *not necessarily* diagonal.

2. The Multichannel AR Model

Multichannel linear prediction is closely related to multichannel auto-regressive (AR) modeling, where we assume that a multichannel signal $\mathbf{x}[n]$ can be represented by the following difference equation:

$$\mathbf{x}[n] = -\mathbf{A}_1^{*T} \cdot \mathbf{x}[n-1] - \mathbf{A}_2^{*T} \cdot \mathbf{x}[n-2] - \ldots - \mathbf{A}_P^{*T} \cdot \mathbf{x}[n-P] + \mathbf{w}[n]. \qquad \text{(IV.12)}$$

Here $\mathbf{A}_i$ are the matrix coefficients and $\mathbf{w}[n]$ is a multichannel white noise process with correlation function

$$\mathbf{r}_w[l] = E\left\{\mathbf{w}[n] \cdot \mathbf{w}^{*T}[n-l]\right\} = \mathbf{\Sigma}_\mathbf{w} \cdot \delta[l]. \qquad \text{(IV.13)}$$

Although the multichannel white noise process is uncorrelated with respect to the lag l, it may be correlated *between channels.* In other words, the covariance matrix $\mathbf{\Sigma}_\mathbf{w}$ is *not necessarily diagonal.*

Linear prediction can also be performed in backward order. When we perform backward linear prediction, we compute the estimate $\hat{\mathbf{x}}[n-P]$ of a multichannel signal as a linear combination of P future values of the signal. The form of the estimate is:

$$\hat{\mathbf{x}}[n-P] = -\mathbf{B}_1^{*T} \cdot \mathbf{x}[n-P+1] - \mathbf{B}_2^{*T} \cdot \mathbf{x}[n-P+2] - \ldots - \mathbf{B}_P^{*T} \cdot \mathbf{x}[n]. \qquad \text{(IV.14)}$$

The AR model matrix coefficients or the linear prediction matrix coefficients can be found from the solution of the *Normal Equations* modified for the multichannel case. As in the single channel case, the Normal Equations can be obtained for forward or backward prediction. We will call $\mathbf{\Sigma}_\varepsilon^f$ the error covariance that results from the forward Normal Equations, and $\mathbf{\Sigma}_\varepsilon^b$ the error covariance that results from the backward Normal Equations. If we call $\mathbf{B}_i$ the coefficients for the backward prediction, we can write the two associated Normal Equations as:

$$\begin{bmatrix} \mathbf{r_x}[0] & \mathbf{r_x}[1] & \cdots & \mathbf{r_x}[P] \\ \mathbf{r_x}[-1] & \mathbf{r_x}[0] & \cdots & \mathbf{r_x}[P-1] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{r_x}[-P] & \mathbf{r_x}[-P+1] & \cdots & \mathbf{r_x}[0] \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I} \\ \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_P \end{bmatrix} = \begin{bmatrix} \mathbf{\Sigma}_\varepsilon^f \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \tag{IV.15}$$

for forward prediction, and

$$\begin{bmatrix} \mathbf{r_x}[0] & \mathbf{r_x}[-1] & \cdots & \mathbf{r_x}[-P] \\ \mathbf{r_x}[1] & \mathbf{r_x}[0] & \cdots & \mathbf{r_x}[-P+1] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{r_x}[P] & \mathbf{r_x}[P-1] & \cdots & \mathbf{r_x}[0] \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I} \\ \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_P \end{bmatrix} = \begin{bmatrix} \mathbf{\Sigma}_\varepsilon^b \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \tag{IV.16}$$

for backward prediction, where $\mathbf{I}$ denotes the identity matrix and $\mathbf{0}$ denotes the zero matrix, both of size $M \times M$. From Equations IV.15 and IV.16 it can be seen that the error covariance matrices $\mathbf{\Sigma}_\varepsilon^f$ and $\mathbf{\Sigma}_\varepsilon^b$ are given by:

$$\mathbf{\Sigma}_\varepsilon^f = \sum_{i=0}^{P} \mathbf{r_x}[i] \cdot \mathbf{A}_i \tag{IV.17}$$

$$\mathbf{\Sigma}_\varepsilon^b = \sum_{i=0}^{P} \mathbf{r_x}[-i] \cdot \mathbf{B}_i \tag{IV.18}$$

where we *define* $\mathbf{A}_0 = \mathbf{B}_0 = \mathbf{I}$.

3. Multichannel Levinson Algorithm

The multichannel Levinson algorithm (Levinson-Wiggins-Robinson [Ref. 14], [Ref. 13]) like its single channel counterpart, provides a recursive method for solving the multichannel Normal Equations. Instead of solving the normal equations directly, the algorithm computes the model parameters of order p from model parameters of order $p-1$. This is done for $p = 0, 1, 2, \ldots, P$. Let us denote the model parameters of order p as:

$$\mathbf{A}_p = \begin{bmatrix} \mathbf{A}_1^{(p)} \\ \vdots \\ \mathbf{A}_p^{(p)} \end{bmatrix}, \quad \mathbf{\Sigma}_p (= \mathbf{\Sigma}_\varepsilon^f), \quad \mathbf{B}_p = \begin{bmatrix} \mathbf{B}_1^{(p)} \\ \vdots \\ \mathbf{B}_p^{(p)} \end{bmatrix}, \quad \mathbf{\Sigma}'_p (= \mathbf{\Sigma}_\varepsilon^b). \tag{IV.19}$$

The algorithm is described by the following procedure:

$$\mathbf{\Delta}_p = \begin{bmatrix} \mathbf{r_x}[1] & \mathbf{r_x}[2] & \cdots & \mathbf{r_x}[p] \end{bmatrix} \cdot \tilde{\mathbf{A}}_{p-1} \tag{IV.20}$$

$$\mathbf{\Gamma}_p = (\mathbf{\Sigma}'_{p-1})^{-1} \cdot \mathbf{\Delta}_p, \qquad \mathbf{\Gamma}'_p = (\mathbf{\Sigma}_{p-1})^{-1} \cdot \mathbf{\Delta}_p^{*T}$$

$$\mathbf{A}_p = \begin{bmatrix} \mathbf{A}_{p-1} \\ \\ -- \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ -- \\ \tilde{\mathbf{B}}_{p-1} \end{bmatrix} \cdot \mathbf{\Gamma}_p, \quad \mathbf{B}_p = \begin{bmatrix} \mathbf{B}_{p-1} \\ \\ -- \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ -- \\ \tilde{\mathbf{A}}_{p-1} \end{bmatrix} \cdot \mathbf{\Gamma}'_p \tag{IV.21}$$

$$\mathbf{\Sigma}_p = \mathbf{\Sigma}_{p-1} \cdot \left(\mathbf{I} - \mathbf{\Gamma}'_p \cdot \mathbf{\Gamma}_p\right), \qquad \mathbf{\Sigma}'_p = \mathbf{\Sigma}'_{p-1} \cdot \left(\mathbf{I} - \mathbf{\Gamma}_p \cdot \mathbf{\Gamma}'_p\right)$$

with initial conditions

$$\mathbf{A}_0 = \mathbf{B}_0 = \mathbf{I} \quad , \quad \mathbf{\Sigma}_0 = \mathbf{\Sigma}'_0 = \mathbf{r_x}[0] \quad , \quad \mathbf{\Delta}_1 = \mathbf{r_x}[1]. \tag{IV.22}$$

The new matrices defined in the algorithm are the $M \times M$ forward and backward reflection coefficient matrices $\mathbf{\Gamma}_p$ and $\mathbf{\Gamma}'_p$ and the partial correlation matrices $\mathbf{\Delta}_p$ and $\mathbf{\Delta}'_p = \mathbf{\Delta}_p^{*T}$. Note that the matrices $\mathbf{A}_p$ and $\mathbf{B}_p$ have increasing dimensions. The algorithm involves *block reversal*, in the sense that the submatrices $\mathbf{A}_i^{(p)}$ are reversed in order within the matrix $\mathbf{A}_p$ without being reversed internally. The algorithm does not present the symmetry of a single channel case. So in general, $\mathbf{B}_p \neq \mathbf{A}_p^{*T}$, $\mathbf{\Gamma}'_p \neq \mathbf{\Gamma}_p^{*T}$ and $\mathbf{\Sigma}'_p \neq \mathbf{\Sigma}_p$.

We can express the prediction error for the forward prediction as

$$\epsilon_p[n] = \sum_{i=0}^{p} \mathbf{A}_i^{(p)^{*T}} \cdot \mathbf{x}[n-i] \tag{IV.23}$$

and for the backward prediction as

$$\epsilon_p^b[n] = \sum_{i=0}^{p} \mathbf{B}_i^{(p)^{*T}} \cdot \mathbf{x}[n-p+i]. \tag{IV.24}$$

With proper substitution in Equations IV.21, we can obtain a recursive form for the prediction error as:

$$\begin{bmatrix} \epsilon_p[n] \\ \\ \epsilon_p^b[n] \end{bmatrix} = \begin{bmatrix} \mathbf{I} & -\mathbf{\Gamma}_p^{*T} \\ \\ -\mathbf{\Gamma}'^{*T}_p & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \epsilon_{p-1}[n] \\ \\ \epsilon_{p-1}^b[n-1] \end{bmatrix}. \tag{IV.25}$$

Equation IV.25 shows that the p^{th} order forward and backward prediction errors can be computed from the corresponding $(p-1)^{th}$ order errors. This is illustrated in Figure 18 where the p^{th} order filter can be formed recursively from the $(p-1)^{th}$ order filter with the addition of a *lattice section.*

This lattice representation is convenient since the parameters of earlier stages do not change as the order of the filter is increased.

C. MULTIRATE LINEAR PREDICTION

In this section we treat the decimated signal as a two-channel random process and we apply the multichannel Levinson algorithm. We start from obtaining a random process $s[n]$ which is zero-mean and stationary in the wide sense. We can form

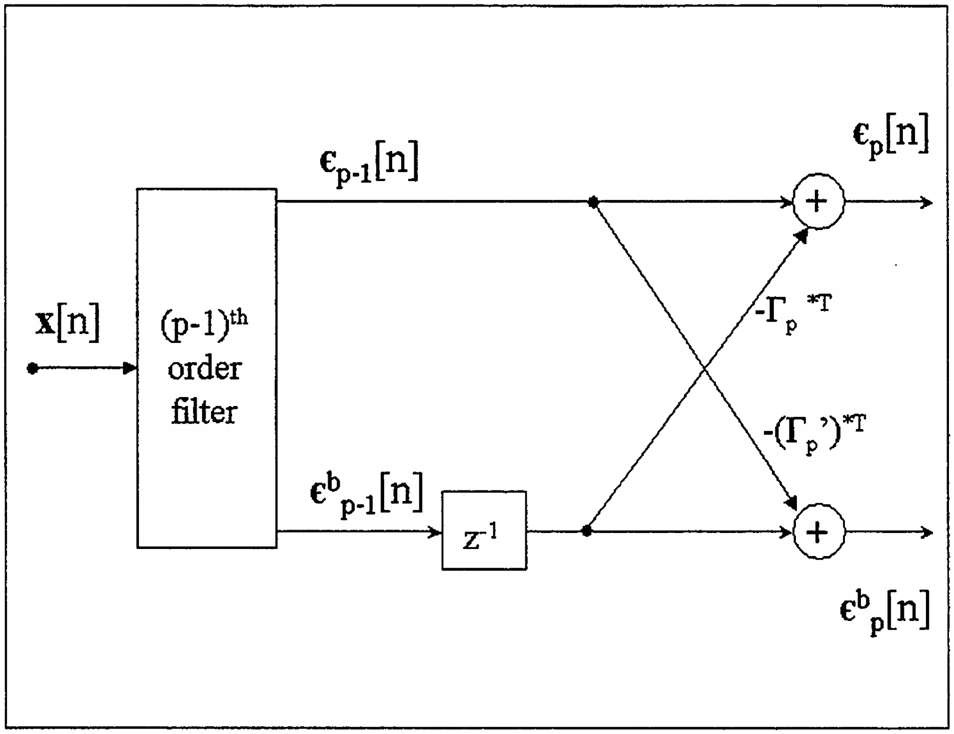

Figure 18. Lattice Section for Multichannel Linear Prediction

a 2-channel signal by obtaining the even and odd samples separately. We define $x_0[n] = s[2n]$ as the process that comes from decimation by a factor of 2 on $s[n]$ with zero time delay, and $x_1[n] = s[2n-1]$ as the process that comes from decimation by a factor of 2 on $s[n]$ with time delay of one. We then define the following signal vector:

$$\mathbf{x}[n] = \begin{bmatrix} x_0[n] \\ x_1[n] \end{bmatrix} = \begin{bmatrix} s[2n] \\ s[2n-1] \end{bmatrix}. \tag{IV.26}$$

The correlation function, can then be computed as:

$$
\begin{aligned}
\mathbf{r}_{\mathbf{x}}[l] &= E\left\{\mathbf{x}[n]\cdot\mathbf{x}^{*T}[n-l]\right\} \\
&= E\left\{\begin{bmatrix} s[2n] \\ s[2n-1] \end{bmatrix}\cdot\begin{bmatrix} s^*[2n-2l] & s^*[2n-1-2l] \end{bmatrix}\right\} \\
&= E\left\{\begin{bmatrix} s[2n]\cdot s^*[2n-2l] & | & s[2n]\cdot s^*[2n-1-2l] \\ s[2n-1]\cdot s^*[2n-2l] & | & s[2n-1]\cdot s^*[2n-1-2l] \end{bmatrix}\right\} \\
&= \begin{bmatrix} R_{ss}[2l] & | & R_{ss}[2l+1] \\ R_{ss}[2l-1] & | & R_{ss}[2l] \end{bmatrix}.
\end{aligned}
\tag{IV.27}
$$

Observe that

$$
\begin{aligned}
\mathbf{r}_{\mathbf{x}}[-l] &= \begin{bmatrix} R_{ss}[2(-l)] & | & R_{ss}[2(-l)+1] \\ R_{ss}[2(-l)-1] & | & R_{ss}[2(-l)] \end{bmatrix} \\
&= \begin{bmatrix} R_{ss}^*[2l] & | & R_{ss}^*[2l-1] \\ R_{ss}^*[2l+1] & | & R_{ss}^*[2l] \end{bmatrix} = \mathbf{r}_{\mathbf{x}}^{*T}[l].
\end{aligned}
\tag{IV.28}
$$

Based on the definition of the multichannel Normal Equations (Equation IV.15), the correlation matrix for forward prediction becomes:

$$
\mathbf{R}^f = \begin{bmatrix}
R_{ss}[0] & R_{ss}[1] & \cdots & \cdots & R_{ss}[2P] & R_{ss}[2P+1] \\
R_{ss}[-1] & R_{ss}[0] & \cdots & \cdots & R_{ss}[2P-1] & R_{ss}[2P] \\
-- & -- & -- & -- & -- & -- \\
R_{ss}[-2] & R_{ss}[-1] & \cdots & \cdots & R_{ss}[2P-2] & R_{ss}[2P-1] \\
R_{ss}[-3] & R_{ss}[-2] & \cdots & \cdots & R_{ss}[2P-3] & R_{ss}[2P-2] \\
-- & -- & -- & -- & -- & -- \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
-- & -- & -- & -- & -- & -- \\
R_{ss}[-2P] & R_{ss}[-2P+1] & \cdots & \cdots & R_{ss}[0] & R_{ss}[1] \\
R_{ss}[-2P-1] & R_{ss}[-2P] & \cdots & \cdots & R_{ss}[-1] & R_{ss}[0]
\end{bmatrix} \tag{IV.29}
$$

It is apparent that $\mathbf{R}^f$ is identical to the matrix that is used for single channel linear prediction of order $2P$. On the other hand, the correlation matrix for backward prediction becomes:

$$\mathbf{R}^b = \left[\begin{array}{cccccc} R_{ss}[0] & R_{ss}[1] & \cdots & \cdots & R_{ss}[-2P] & R_{ss}[-2P+1] \\ R_{ss}[-1] & R_{ss}[0] & \cdots & \cdots & R_{ss}[-2P-1] & R_{ss}[-2P] \\ -- & -- & -- & -- & -- & -- \\ R_{ss}[2] & R_{ss}[3] & \cdots & \cdots & R_{ss}[-2P+2] & R_{ss}[-2P+3] \\ R_{ss}[1] & R_{ss}[2] & \cdots & \cdots & R_{ss}[-2P+1] & R_{ss}[-2P-2] \\ -- & -- & -- & -- & -- & -- \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ -- & -- & -- & -- & -- & -- \\ R_{ss}[2P] & R_{ss}[2P+1] & \cdots & \cdots & R_{ss}[0] & R_{ss}[1] \\ R_{ss}[2P-1] & R_{ss}[2P] & \cdots & \cdots & R_{ss}[-1] & R_{ss}[0] \end{array}\right] \tag{IV.30}$$

A closer look at Equations IV.26, IV.27, IV.29 and IV.30 reveals the fact that if we reverse the signal vector in Equation IV.26, the matrices $\mathbf{R}^f$ and $\mathbf{R}^b$ are interchanged. The last observation indicates that the choice for the order of the channels within the signal vector leads to equivalent results, a fact that is in line with what we should expect.

1. Linear Prediction Output

A possible way to check the output of linear prediction on the multirate process, is to try to form the prediction error. Based on Equation IV.10, the prediction error in this case will be:

$$\epsilon[n] = \mathbf{x}[n] - \hat{\mathbf{x}}[n] = \sum_{i=0}^{P} \mathbf{A}_i^{*T} \cdot \mathbf{x}[n-i] \tag{IV.31}$$

or equvalently,

$$\begin{aligned}
\begin{bmatrix} \varepsilon_0[n] \\ \varepsilon_1[n] \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s[2n] \\ s[2n-1] \end{bmatrix} + \textstyle\sum_{i=1}^{P} \mathbf{A}_i^{*T} \cdot \begin{bmatrix} s[2n-2i] \\ s[2n-2i-1] \end{bmatrix} \\
&= \begin{bmatrix} s[2n] \\ s[2n-1] \end{bmatrix} + \begin{bmatrix} A_1^{(00)*} & A_1^{(10)*} \\ A_1^{(01)*} & A_1^{(11)*} \end{bmatrix} \cdot \begin{bmatrix} s[2n-2] \\ s[2n-3] \end{bmatrix} + \ldots \qquad \text{(IV.32)} \\
&= \begin{bmatrix} s[2n] + A_1^{(00)*} \cdot s[2n-2] + A_1^{(10)*} \cdot s[2n-3] + \ldots \\ s[2n-1] + A_1^{(01)*} \cdot s[2n-2] + A_1^{(11)*} \cdot s[2n-3] + \ldots \end{bmatrix}.
\end{aligned}$$

Equation IV.32 provides an indication that at least one of the components of $\epsilon[n]$ (in particular $\varepsilon_1[n]$) is essentially the same as the error $\varepsilon[n]$ that is the output of linear prediction in the single channel case on the same signal. This fact is also confirmed by the numerical results.

2. Numerical Results

For the purpose of performing multichannel linear prediction on the multirate random process, a MATLAB function was created with the name ChanLevin.m . The main purpose of this function was to implement the multirate Levinson recursion and show the results for $\mathbf{A}_p$, $\mathbf{B}_p$, $\mathbf{\Gamma}_p$, $\mathbf{\Gamma}'_p$, $\mathbf{\Sigma}_p$ and $\mathbf{\Sigma}'_p$ for comparison and further analysis. Two autocorrelation functions were created for two AR models of orders two and three, (two and three poles respectively). The idea behind this was to exclude any chance that the observed experimental properties would be accidental. The first model was actually the same that was used in the Multirate filter experiments (shown in Figure 10); with generating noise variance $\sigma_w^2 = 1$ and poles at 0.5 and 0.8. For the second model the generating noise variance was also $\sigma_w^2 = 1$ and the poles were at 0.4, 0.5 and 0.7. For both cases the linear prediction was performed for order $P = 2$. Since the underlying model is of order 2 and the theoretical autocorrelation function was also computed for these results, linear prediction of higher order would not be necessary.

The 2^{nd} order AR model is described by the difference equation

$$s[n] = 1.3s[n-1] - 0.4s[n-2] + w[n] \tag{IV.33}$$

where $\sigma_w^2 = 1$. This corresponds to a prediction error filter of the form $A(z) = 1 - 1.3z^{-1} + 0.4z^{-2}$. The autocorrelation function terms for the process are:

$$\begin{gathered} R_{ss}[0] = 8.6420 \quad R_{ss}[1] = 8.0247 \quad R_{ss}[2] = 6.9753 \\ R_{ss}[3] = 5.8580 \quad R_{ss}[4] = 4.8253 \quad R_{ss}[5] = 3.9297. \end{gathered} \tag{IV.34}$$

Therefore, the correlation matrix blocks for the multichannel signal are given by:

$$\begin{gathered} \mathbf{r_x}[0] = \begin{bmatrix} 8.6420 & 8.0247 \\ 8.0247 & 8.6420 \end{bmatrix} \quad \mathbf{r_x}[1] = \begin{bmatrix} 6.9753 & 5.8580 \\ 8.0247 & 6.9753 \end{bmatrix} \\ \mathbf{r_x}[2] = \begin{bmatrix} 4.8253 & 3.9297 \\ 5.8580 & 4.8253 \end{bmatrix}. \end{gathered} \tag{IV.35}$$

Finally, the results were:

$$\begin{bmatrix} \mathbf{\Gamma}_1 \\ -- \\ \mathbf{\Gamma}_2 \end{bmatrix} = \begin{bmatrix} 1.29 & 1.30 \\ -0.52 & -0.40 \\ -- & -- \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{\Gamma}'_1 \\ -- \\ \mathbf{\Gamma}'_2 \end{bmatrix} = \begin{bmatrix} -0.40 & -0.52 \\ 1.30 & 1.29 \\ -- & -- \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{IV.36}$$

$$\mathbf{A}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -- & -- \\ -1.29 & -1.30 \\ 0.52 & 0.40 \\ -- & -- \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -- & -- \\ 0.40 & 0.52 \\ -1.30 & -1.29 \\ -- & -- \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad \text{(IV.37)}$$

$$\mathbf{\Sigma}_2 = \begin{bmatrix} 2.69 & 1.30 \\ 1.30 & 1.00 \end{bmatrix}, \quad \mathbf{\Sigma}_2' = \begin{bmatrix} 1.00 & 1.30 \\ 1.30 & 2.69 \end{bmatrix}. \qquad \text{(IV.38)}$$

The 3^{rd} order AR model is described by the difference equation:

$$s[n] = 1.6s[n-1] - 0.83s[n-2] + 0.14s[n-3] + w[n] \qquad \text{(IV.39)}$$

where $\sigma_w^2 = 1$. This corresponds to a prediction error filter of the form $A(z) = 1 - 1.6z^{-1} + 0.83z^{-2} - 0.14z^{-3}$. The autocorrelation function terms for the process are:

$$\begin{array}{lll} R_{ss}[0] = 13.1876 & R_{ss}[1] = 12.3347 & R_{ss}[2] = 10.5167 \\ R_{ss}[3] = 8.4351 & R_{ss}[4] = 6.4942 & R_{ss}[5] = 4.8619. \end{array} \qquad \text{(IV.40)}$$

Therefore, the correlation matrix blocks for the multichannel signal are given by:

$$\mathbf{r}_{\mathbf{x}}[0] = \begin{bmatrix} 13.1876 & 12.3347 \\ 12.3347 & 13.1876 \end{bmatrix} \quad \mathbf{r}_{\mathbf{x}}[1] = \begin{bmatrix} 10.5167 & 8.4351 \\ 12.3347 & 10.5167 \end{bmatrix}$$
$$\mathbf{r}_{\mathbf{x}}[2] = \begin{bmatrix} 6.4942 & 4.8619 \\ 8.4351 & 6.4942 \end{bmatrix}. \tag{IV.41}$$

The obtained results were:

$$\begin{bmatrix} \mathbf{\Gamma}_1 \\ -- \\ \mathbf{\Gamma}_2 \end{bmatrix} = \begin{bmatrix} 1.5915 & 1.5135 \\ -0.8490 & -0.6181 \\ -- & -- \\ 0.2240 & 0.1400 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{\Gamma}'_1 \\ -- \\ \mathbf{\Gamma}'_2 \end{bmatrix} = \begin{bmatrix} -0.6181 & -0.8490 \\ 1.5135 & 1.5915 \\ -- & -- \\ 0 & 0 \\ 0.1400 & 0.2240 \end{bmatrix} \tag{IV.42}$$

$$\mathbf{A}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -- & -- \\ -1.7300 & -1.6000 \\ 1.1880 & 0.8300 \\ -- & -- \\ -0.2240 & -0.1400 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -- & -- \\ 0.8300 & 1.1880 \\ -1.6000 & -1.7300 \\ -- & -- \\ 0 & 0 \\ -0.1400 & -0.2240 \end{bmatrix} \tag{IV.43}$$

$$\mathbf{\Sigma}_2 = \begin{bmatrix} 3.56 & 1.60 \\ 1.60 & 1.00 \end{bmatrix}, \quad \mathbf{\Sigma}'_2 = \begin{bmatrix} 1.00 & 1.60 \\ 1.60 & 3.56 \end{bmatrix}. \tag{IV.44}$$

From Equations IV.36 through IV.44, we can make the following observations:

1. The error covariance for both forward and backward linear prediction is not diagonal. In addition, the diagonal elements are not equal. However, the cross-covariance elements are equal as they must be, since a covariance matrix is always symmetric.

2. *All* 2×2 components *are the reverses* of their counterparts. In other words:

$$\mathbf{B}_i^{(p)} = \tilde{\mathbf{A}}_i^{(p)}, \quad \mathbf{\Gamma}'_p = \tilde{\mathbf{\Gamma}}_p, \quad \mathbf{\Sigma}'_p = \tilde{\mathbf{\Sigma}}_p. \tag{IV.45}$$

   This makes sense since the signal vector is consisted of adjacent elements of the same random process.

3. The observation made based on Equation IV.32 proved to be correct. The second column of $\mathbf{A}_p$ (the one that corresponds to $\epsilon_1[n]$) has the same coefficients as the generating polynomial $A(z)$. In addition, the first column of $\mathbf{B}_p$ has also the same coefficients as the generating polynomial $A(z)$ in reverse order.

4. If we examine $\mathbf{A}_p$ and $\mathbf{B}_p$ columnwise, the number of non-zero elements are exactly the same as in the generating polynomial. Therefore, there is no computational loss or gain from the multichannel prediction. However, the number of matrix coefficients becomes half the selected order for large P.

D. DIAGONALIZING THE ERROR COVARIANCE

Despite the success in the implementation of multichannel linear prediction in the specific setup of even and odd samples of a random process and the remarkable symmetry that makes the particular use of multichannel prediction equivalent to its single channel counterpart, there is one problem that still remains unsolved. It is the fact that the error covariance still remains non-diagonal. If we recall the polyphase representation and the polyphase matrix from the beginning of the chapter, we would like to invert the model and design a whitening filter. In this way, we would obtain the innovation from the process by obtaining even and odd samples from two uncorrelated innovations.

In this section we consider the application of suitable transformations that would diagonalize the error covariance matrix. We can consider two possible generic types of decompositions:

1. *Eigenvalue decomposition*: The eigenvalue problem is to determine the nontrivial solutions of the equation $\mathbf{A} \cdot \mathbf{x} = \lambda \cdot \mathbf{x}$. If $\mathbf{A}$ is an $N \times N$ matrix, the full solution of the problem includes N values that satisfy the equation called eigenvalues and arranged in a diagonal matrix $\mathbf{\Lambda}$. The corresponding vectors of $\mathbf{x}$ are called right eigenvectors and can be arranged as the columns of a matrix $\mathbf{V}$. If $\mathbf{A}$ is Hermitian symmetric (conjugate symmetric), the eigenvectors are always linearly independent and the eigenvector matrix $\mathbf{V}$ can be used to perform a similarity transformation, leading to the result $\mathbf{V}^{*T} \cdot \mathbf{A} \cdot \mathbf{V} = \mathbf{\Lambda}$.

2. *Cholesky factorization*: The Cholesky factorization [Ref. 10: pp.195] is another method that can be used to diagonalize a matrix. If $\mathbf{A}$ is Hermitian symmetric, it can be factorized as $\mathbf{A} = \mathbf{L} \cdot \mathbf{D} \cdot \mathbf{L}^{*T}$, where $\mathbf{L}$ is lower triangular and invertible and $\mathbf{D}$ is diagonal. We can perform a similarity transformation on $\mathbf{A}$, leading to the result $\mathbf{L}^{-1} \cdot \mathbf{A} \cdot (\mathbf{L}^{*T})^{-1} = \mathbf{D}$.

1. Diagonalizing $\mathbf{\Sigma}_\varepsilon^f$ with Eigenvalue Decomposition

Since the error covariance matrix $\mathbf{\Sigma}_\varepsilon^f$ is symmetric, it can be decomposed in terms as $\mathbf{\Sigma}_\varepsilon^f = \mathbf{V} \cdot \mathbf{\Lambda} \cdot \mathbf{V}^{*T}$, where $\mathbf{V}$ is the matrix of right eigenvectors and $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues. From Equation IV.17 we can write:

$$\mathbf{\Sigma}_\varepsilon^f = \mathbf{V} \cdot \mathbf{\Lambda} \cdot \mathbf{V}^{*T} = \sum_{i=0}^{P} \mathbf{r}_{\mathbf{x}}[i] \cdot \mathbf{A}_i \tag{IV.46}$$

and therefore

$$\mathbf{\Lambda} = \sum_{i=0}^{P} \mathbf{V}^{*T} \cdot \mathbf{r}_{\mathbf{x}}[i] \cdot \mathbf{A}_i \cdot \mathbf{V} \tag{IV.47}$$

where we have used the fact that $\mathbf{V}^{*T} \cdot \mathbf{V} = \mathbf{V} \cdot \mathbf{V}^{*T} = \mathbf{I}$. If we define a new signal vector $\mathbf{y}[n] = \mathbf{V}^{*T} \cdot \mathbf{x}[n]$, then its correlation function becomes:

$$\begin{aligned} \mathbf{r}_{\mathbf{y}}[l] &= E\left\{\mathbf{y}[n] \cdot \mathbf{y}^{*T}[n-l]\right\} = E\left\{\mathbf{V}^{*T} \cdot \mathbf{x}[n] \cdot \mathbf{x}^{*T}[n-l] \cdot \mathbf{V}\right\} \\ &= \mathbf{V}^{*T} \cdot \mathbf{r}_{\mathbf{x}}[l] \cdot \mathbf{V}. \end{aligned} \tag{IV.48}$$

Furthermore, we can define the transformed coefficient matrices as

$$\mathbf{C}_i = \mathbf{V}^{*T} \cdot \mathbf{A}_i \cdot \mathbf{V} \quad , \quad i = 0, 1, 2, \ldots, P \tag{IV.49}$$

and write again Equation IV.47 as:

$$\mathbf{\Lambda} = \sum_{i=0}^{P} \mathbf{r_y}[i] \cdot \mathbf{C}_i. \qquad \text{(IV.50)}$$

This also means that the error signal vector for the prediction of $\mathbf{y}[n]$ can be written as:

$$\begin{aligned} \epsilon'[n] &= \textstyle\sum_{i=0}^{P} \mathbf{C}_i^{*T} \cdot \mathbf{y}[n-i] = \sum_{i=0}^{P} \mathbf{V}^{*T} \cdot \mathbf{A}_i^{*T} \cdot \mathbf{V} \cdot \mathbf{V}^{*T} \cdot \mathbf{x}[n-i] \\ &= \mathbf{V}^{*T} \cdot \textstyle\sum_{i=0}^{P} \mathbf{A}_i^{*T} \cdot \mathbf{x}[n-i] = \mathbf{V}^{*T} \cdot \epsilon[n]. \end{aligned} \qquad \text{(IV.51)}$$

In conclusion, the transformation $\mathbf{y}[n] = \mathbf{V}^{*T} \cdot \mathbf{x}[n]$ yields the prediction matrices $\mathbf{C}_i = \mathbf{V}^{*T} \cdot \mathbf{A}_i \cdot \mathbf{V}$, and a forward prediction error vector with uncorrelated channels.

Let us now consider what happens with backward prediction. We have seen in the numerical examples (assuming that the signal on which linear prediction is performed comes from an AR model) that $\mathbf{\Sigma}_\varepsilon^b$ is the reverse of $\mathbf{\Sigma}_\varepsilon^f$. It can easily be shown therefore that the eigenvector matrices for $\mathbf{\Sigma}_\varepsilon^f$ and $\mathbf{\Sigma}_\varepsilon^b$ are $\mathbf{V}$ and $\mathbf{V}^{*T}$ respectively. That means that we can determine a similar transformation for the backward prediction, defined as: $\mathbf{y}^b[n] = \mathbf{V} \cdot \mathbf{x}[n]$ with $\mathbf{C}_i^b = \mathbf{V} \cdot \mathbf{A}_i \cdot \mathbf{V}^{*T}$. This transformation yields a backward prediction error with diagonal error covariance matrix. As a result of these considerations we see that we can apply a transformation to diagonalize *either* the forward or the backward error covariance but (unless $\mathbf{V}^{*T} = \mathbf{V}$) we cannot diagonalize both simultaneously without changing the scale of the errors.

2. Diagonalizing $\mathbf{\Sigma}_\varepsilon^f$ with Cholesky Factorization

By an argument similar to the previous section, we can express $\mathbf{\Sigma}_\varepsilon^f$ using the Cholesky factorization as:

$$\mathbf{\Sigma}_\epsilon^f = \mathbf{L} \cdot \mathbf{D} \cdot \mathbf{L}^{*T} = \sum_{i=0}^{P} \mathbf{r}_\mathbf{x}[i] \cdot \mathbf{A}_i. \tag{IV.52}$$

By pre-multiplication and post-multiplication with $\mathbf{L}^{-1}$ and $(\mathbf{L}^{*T})^{-1}$ respectively, we obtain

$$\mathbf{D} = \sum_{i=0}^{P} \mathbf{L}^{-1} \cdot \mathbf{r}_\mathbf{x}[i] \cdot \mathbf{A}_i \cdot (\mathbf{L}^{*T})^{-1}. \tag{IV.53}$$

We again define a new signal vector $\mathbf{y}[n] = \mathbf{L}^{-1} \cdot \mathbf{x}[n]$, with correlation function:

$$\begin{aligned} \mathbf{r}_\mathbf{y}[l] &= E\left\{\mathbf{y}[n] \cdot \mathbf{y}^{*T}[n-l]\right\} = E\left\{\mathbf{L}^{-1} \cdot \mathbf{x}[n] \cdot \mathbf{x}^{*T}[n-l] \cdot (\mathbf{L}^{*T})^{-1}\right\} \\ &= \mathbf{L}^{-1} \cdot \mathbf{r}_\mathbf{x}[l] \cdot (\mathbf{L}^{*T})^{-1} \end{aligned} \tag{IV.54}$$

and by transforming the coefficient matrices to

$$\mathbf{C}_i = \mathbf{L}^{*T} \cdot \mathbf{A}_i \cdot (\mathbf{L}^{*T})^{-1} \quad , \quad i = 0, 1, 2, \ldots, P \tag{IV.55}$$

we can write Equation IV.53 as:

$$\mathbf{D} = \sum_{i=0}^{P} \mathbf{r}_\mathbf{y}[i] \cdot \mathbf{C}_i. \tag{IV.56}$$

Finally, the error signal vector for the prediction of $\mathbf{y}[n]$ can be written as:

$$\begin{aligned} \epsilon'[n] &= \textstyle\sum_{i=0}^{P} \mathbf{C}_i^{*T} \cdot \mathbf{y}[n-i] = \sum_{i=0}^{P} \mathbf{L}^{-1} \cdot \mathbf{A}_i^{*T} \cdot \mathbf{L} \cdot \mathbf{L}^{-1} \cdot \mathbf{x}[n-i] \\ &= \mathbf{L}^{-1} \cdot \textstyle\sum_{i=0}^{P} \mathbf{A}_i^{*T} \cdot \mathbf{x}[n-i] = \mathbf{L}^{-1} \cdot \epsilon[n]. \end{aligned} \tag{IV.57}$$

Thus, with the use of the transformation $\mathbf{y}[n] = \mathbf{L}^{-1} \cdot \mathbf{x}[n]$, accompanied with $\mathbf{C}_i = \mathbf{L}^{*T} \cdot \mathbf{A}_i \cdot (\mathbf{L}^{*T})^{-1}$, we effectively make the signal vector produce a diagonal error covariance matrix under forward linear prediction. Under backward prediction the case is similar to that of eigenvalue decomposition, and the corresponding transformation for the backward prediction is given by: $\mathbf{y}^b[n] = (\mathbf{L}^{*T})^{-1} \cdot \mathbf{x}[n]$ with coefficient matrices $\mathbf{C}_i^b = \mathbf{L}^{-1} \cdot \mathbf{A}_i \cdot \mathbf{L}$. The diagonal matrix $\mathbf{D}$ has all the diagonal elements equal. As in the previous case, the forward and backward predictions have two different diagonalizing transformations.

3. Numerical Results

In order to obtain results with the use of eigenvalue decomposition and Cholesky factorization, each of the 2×2 correlation function blocks (see Equation IV.41) were transformed according to the Equations IV.48 and IV.54. Afterwards, the multichannel linear prediction was performed with the function ChanLevin.m on the resulting correlation matrix. This time, only one AR model was used, the one that is shown in Figure 10 with generating noise variance $\sigma_w^2 = 1$ and poles at 0.5 and 0.8. The linear prediction was performed for order $P = 2$.

With the use of eigenvalue decomposition the results were:

$$\begin{bmatrix} \mathbf{\Gamma}_1 \\ -- \\ \mathbf{\Gamma}_2 \end{bmatrix} = \begin{bmatrix} 1.2325 & 0.4141 \\ -1.4059 & -0.3425 \\ -- & -- \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{\Gamma}'_1 \\ -- \\ \mathbf{\Gamma}'_2 \end{bmatrix} = \begin{bmatrix} 0.3115 & 0.0110 \\ 1.8310 & 0.5785 \\ -- & -- \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad \text{(IV.58)}$$

$$\mathbf{A}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -- & -- \\ -1.2325 & -0.4141 \\ 1.4059 & 0.3425 \\ -- & -- \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -- & -- \\ -0.3115 & -0.0110 \\ -1.8310 & -0.5785 \\ -- & -- \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{IV.59}$$

$$\mathbf{\Sigma}_2 = \begin{bmatrix} 3.3955 & 0 \\ 0 & 0.2945 \end{bmatrix}, \quad \mathbf{\Sigma}'_2 = \begin{bmatrix} 2.4745 & 1.4170 \\ 1.4170 & 1.2155 \end{bmatrix}. \tag{IV.60}$$

With the use of Cholesky factorization the results were as follows:

$$\begin{bmatrix} \mathbf{\Gamma}_1 \\ -- \\ \mathbf{\Gamma}_2 \end{bmatrix} = \begin{bmatrix} 1.0387 & 1.6267 \\ -0.1933 & -0.1487 \\ -- & -- \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{\Gamma}'_1 \\ -- \\ \mathbf{\Gamma}'_2 \end{bmatrix} = \begin{bmatrix} 0.2283 & -0.0185 \\ 0.4833 & 0.6617 \\ -- & -- \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{IV.61}$$

$$\mathbf{A}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -- & -- \\ -1.0387 & -1.6267 \\ 0.1933 & 0.1487 \\ -- & -- \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -- & -- \\ -0.2283 & 0.0185 \\ -0.4833 & -0.6617 \\ -- & -- \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad \text{(IV.62)}$$

$$\mathbf{\Sigma}_2 = \begin{bmatrix} 1.0000 & 0.0000 \\ 0.0000 & 1.0000 \end{bmatrix}, \quad \mathbf{\Sigma}_2' = \begin{bmatrix} 0.3717 & 0.8167 \\ 0.8167 & 4.4844 \end{bmatrix}. \qquad \text{(IV.63)}$$

From a close examination of Equations IV.58 through IV.63, we can make the following observations:

1. Both transformations yield the expected results in terms of the error covariance $\mathbf{\Sigma}_2$ for linear prediction. For the case of eigenvalue decomposition the error covariance matrix (Equation IV.60) was equal to the eigenvalue matrix. For the case of Cholesky factorization (Equation IV.63) the covariance of error was the *identity matrix.*
2. There is no symmetry between the resulting 2×2 components in forward and backward prediction. As a consequence, $\mathbf{\Sigma}_\varepsilon^f$ is different than $\mathbf{\Sigma}_\varepsilon^b$, and $\mathbf{\Sigma}_\varepsilon^b$ is not necessarily diagonal.
3. Neither the elements of $\mathbf{A}_p$ nor the elements of $\mathbf{B}_p$ include the generating polynomial $A(z)$, neither in the forward nor in the reverse order.

4. Simulation Results

A simulation was performed in order to check the performance of linear prediction on data. The AR model that was used for data generation was the one shown in Figure 10 with generating noise variance $\sigma_w^2 = 1$ and poles at 0.5 and 0.8. The signal

$s[n]$ formed two channels according to Equation IV.26 as $\mathbf{x}[n]$. A linear predictor of order $P = 2$ was designed for $\mathbf{x}[n]$. The residual error sequence $\epsilon[n]$ was transformed by $\mathbf{L}^{-1}$ according to Equation IV.57 (Cholesky factorization), producing $\epsilon'[n]$.

For both $\epsilon'_0[n]$ and $\epsilon'_1[n]$ the autocorrelation function was computed, in order to check the theoretical prediction that they are orthogonal sequences. Also, their cross-correlation was computed in order to check the prediction that they are uncorrelated. The obtained results are presented in Figure 19, from we come to the conclusion that the implementation of Cholesky factorization transformed the two channels of the residual $\epsilon'[n]$ to *orthonormal* sequences.

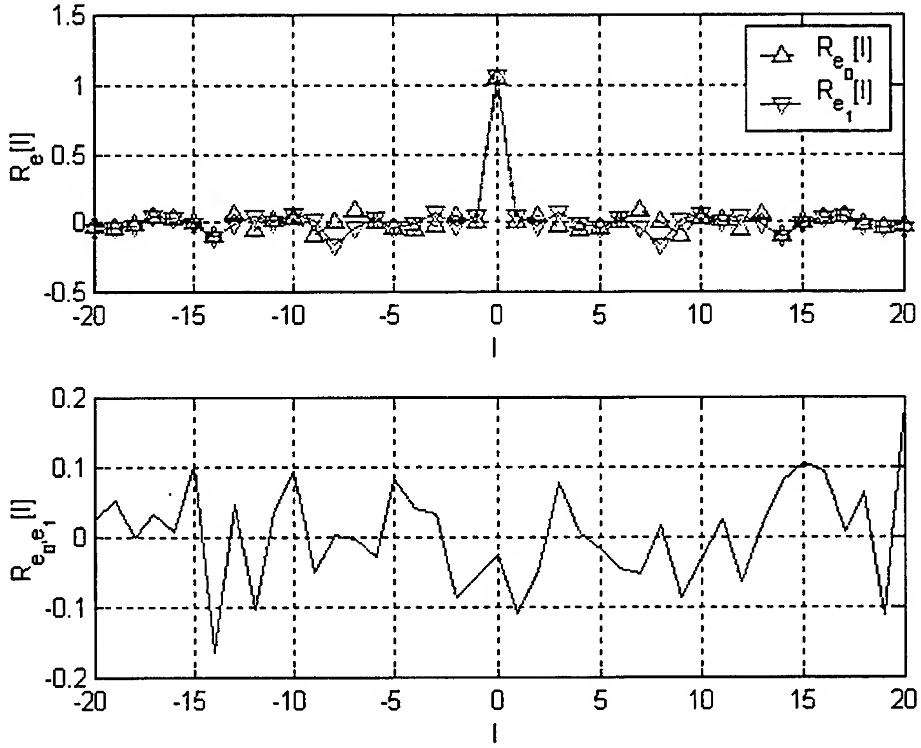

Figure 19. Correlation Functions Between the Channels of $\epsilon'[n]$

The residual $\epsilon[n]$ was filtered from the inverse lattice based on the linear prediction of $\mathbf{x}[n]$ and expanded in order to check for perfect reconstruction of $s[n]$. The obtained estimate $\hat{s}[n]$ presented in Figure 20 is actually $s[n]$ delayed by two samples, since it comes from a two step ahead prediction. The figure confirms there is perfect signal reconstruction.

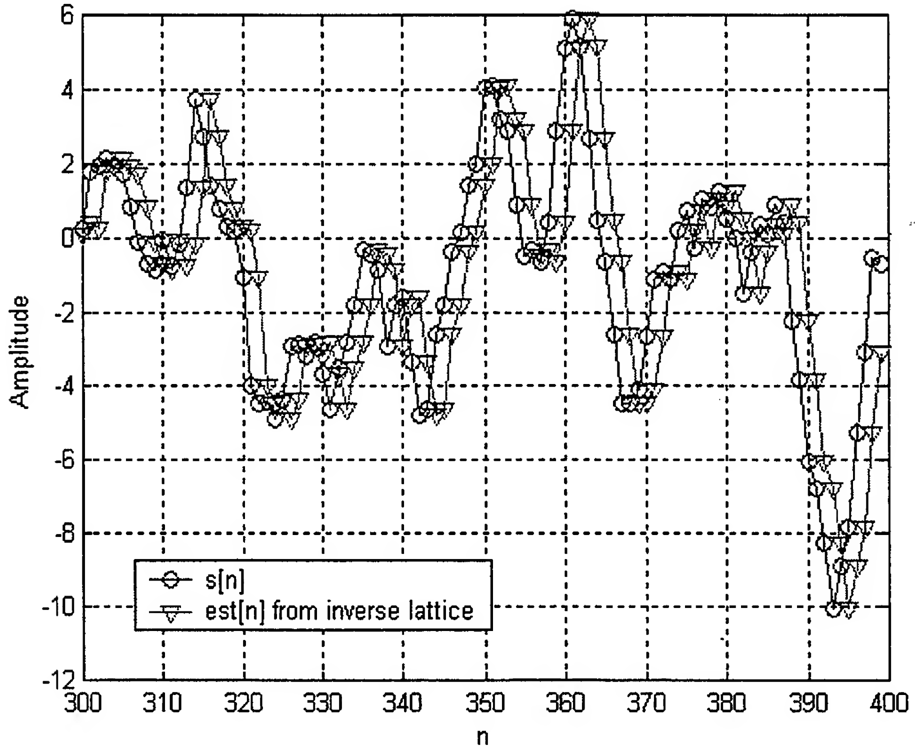

Figure 20. Estimated Signal $\hat{s}[n]$ and $s[n]$ from Inverse Lattice on $\epsilon[n]$

The residual $\epsilon'[n]$ was filtered through the inverse lattice, based on linear prediction on $\mathbf{y}[n] = \mathbf{L}^{-1} \cdot \mathbf{x}[n]$ (Cholesky factorization), and expanded in order to check possible correlation with $s[n]$. The result, presented in Figure 21, shows that although the obtained estimate presents similar statistics, it is not an approximation of $s[n]$.

We have shown that the vector $\epsilon'[n]$ is stated as $\epsilon'[n] = \mathbf{L}^{-1} \cdot \epsilon[n]$, therefore we can invert the situation and write $\epsilon[n] = \mathbf{L} \cdot \epsilon'[n]$. Since $\mathbf{L}$ is a lower triangular matrix, it can be written as:

$$\mathbf{L} = \begin{bmatrix} l_{00} & 0 \\ l_{10} & l_{11} \end{bmatrix}. \tag{IV.64}$$

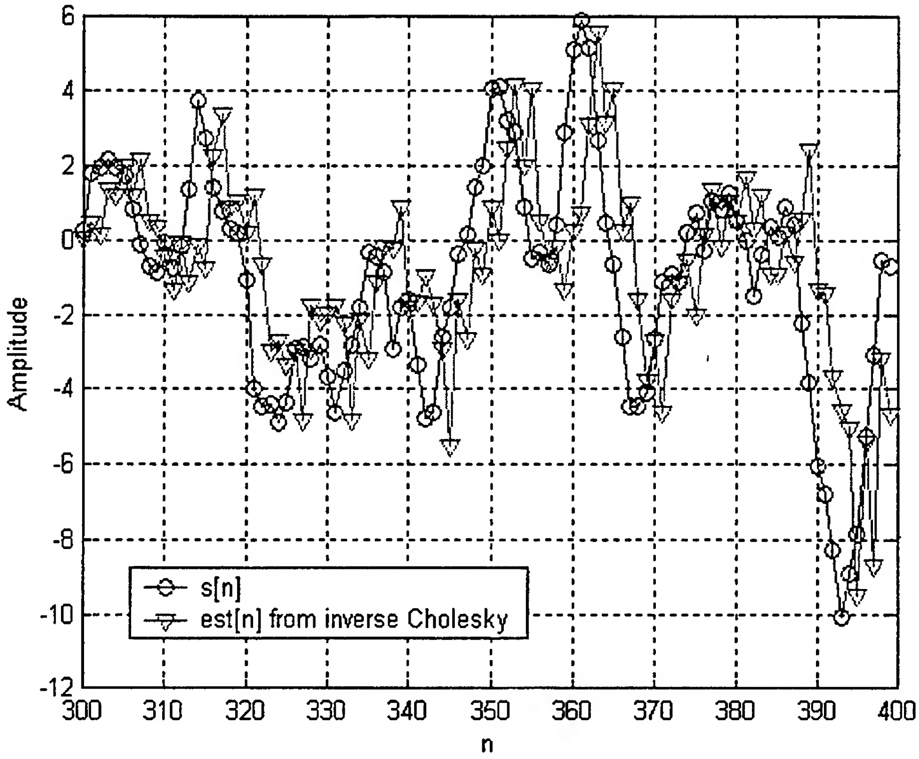

Figure 21. Estimated Signal $\hat{s}[n]$ and $s[n]$ Directly from Cholesky Factorization

The significance of the Cholesky factorization approach comes from the fact that the elements of $\epsilon[n]$ can be written as:

$$\begin{aligned} \epsilon_0[n] &= \; l_{00} \cdot \epsilon_0'[n] \\ \epsilon_1[n] &= \; l_{10} \cdot \epsilon_0'[n] + l_{11} \cdot \epsilon_1'[n] \end{aligned} \qquad \text{(IV.65)}$$

with $\epsilon_0'[n]$ and $\epsilon_1'[n]$ two orthogonal, uncorrelated sequences. As a consequence, given only $\epsilon_0'[n]$, we can reconstruct an approximation of the original signal from the innovation $\epsilon_0[n]$. The additional information needed to reconstruct $\epsilon_1[n]$, and therefore the signal $s[n]$, is the term $\epsilon_1'[n]$ which is uncorrelated to $\epsilon_0'[n]$. This is described in Figure 22, which represents $\epsilon_0'[n]$ and $\epsilon_1'[n]$ as two orthogonal (uncorrelated) vectors in a 2-D vector space.

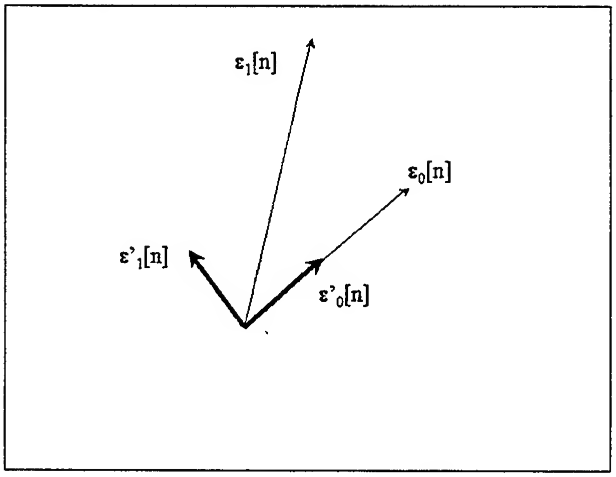

Figure 22. Orthonormal Vector Representation of $\epsilon'[n]$

The importance of the first channel of the residual $\epsilon_0[n]$ becomes evident from the fact that it can be used to produce an approximation of $s[n]$. We can use the principles of polyphase representation as they were stated in the first section of this Chapter and decompose the transfer function $H(z)$ of the AR model into $H_0(z)$ and $H_1(z)$, leading to:

$$H(z) = \frac{1}{(1 - 0.5 \cdot z^{-1}) \cdot (1 - 0.8 \cdot z^{-1})} \tag{IV.66}$$

$$H_0(z) = \frac{1 + (0.5 \cdot 0.8) \cdot z^{-1}}{(1 - 0.25 \cdot z^{-1}) \cdot (1 - 0.64 \cdot z^{-1})} \tag{IV.67}$$

$$H_1(z) = \frac{(0.5 + 0.8)}{(1 - 0.25 \cdot z^{-1}) \cdot (1 - 0.64 \cdot z^{-1})} \tag{IV.68}$$

The first channel of the residual $\epsilon_0[n]$, was filtered from $H_0(z)$ and the result was interpolated and compared with $s[n]$. The result is presented in Figure 23, where the estimate approximates the original signal $s[n]$. However, the same operation can be performed with $\epsilon'_0[n]$ after multiplying it with the scalar quantity l_{00}.

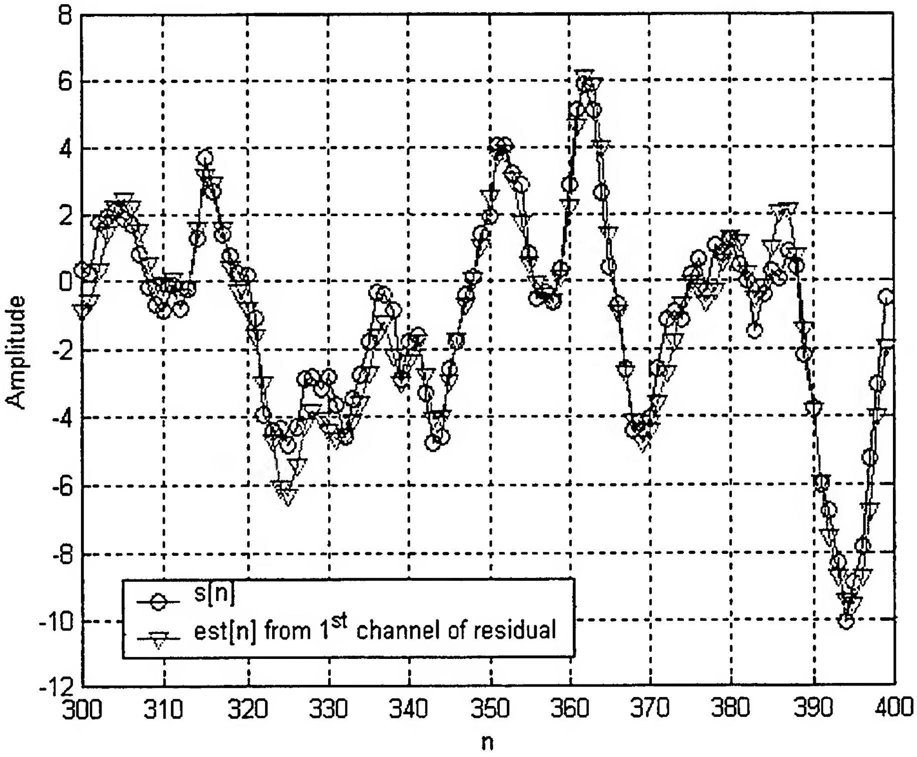

Figure 23. Signal $s[n]$ and $\hat{s}[n]$ from Filtering and Interpolation on $\epsilon_0[n]$

The importance of the last observation comes from the fact that *we only need half of the samples* of the residual to reconstruct an approximation of $s[n]$, based on part of the prediction coefficients (the second column of $\mathbf{A}_p$ as it appears in Equation IV.37). In addition, the transformation of $\epsilon[n]$ from $\mathbf{L}^{-1}$ produces two orthonormal sequences ($\epsilon_0'[n]$ and $\epsilon_1'[n]$), from which we only need the first to obtain $\epsilon_0[n]$. The last result provides to $\epsilon_0'[n]$ a functionality similar to that of the approximation coefficients in Wavelet decomposition (see [Ref. 1]).

V. CONCLUSIONS

In this research we introduced multirate, multiresolution techniques to process random signals. In particular we focused on two objectives: estimation of a signal from a number of observation sequences at different sampling rates, and modeling of a signal presented in multiple channels formed from successive samples. The purpose of the latter was to design a set of filters to whiten the signal.

For the estimation problem, the multirate Wiener filter was designed as an extension of the single rate Wiener filter with two observation sequences, one at high sampling rate and the other at low sampling rate. The technique is general enough to be extended to multiple observation sequences, each at a sampling rate that is a fraction of the basic sampling rate. The simulations that accompany the theoretical derivations, confirm the improved performance of the proposed technique. In addition, the proposed technique is extendable to fields such as Kalman filtering, adaptive filtering and also 2-D filtering and estimation.

The signal modeling problem employed multichannel linear prediction. The motivation at the basis of the multichannel predictor is the representation of a signal by two systems and two uncorrelated white noise sequences. The two systems can be easily related either to the lower resolution and the higher resolution components of the signal, or to the even and odd samples of the signal itself. The fact that the two innovation sequences are white and uncorrelated is attractive in the sense that they represent the minimum amount of information (in the sense of minimum variance) required to reconstruct the signal.

A possible application of this signal modeling is in speech coding using Linear Prediction, as shown in Figure 24. The overall information can be coded at a lower resolution, as an approximation of the original signal, and the details encoded in an uncorrelated innovation. The need to transmit the model parameters as side information represents a minimal overhead with respect to the overall amount of data to be transmitted.

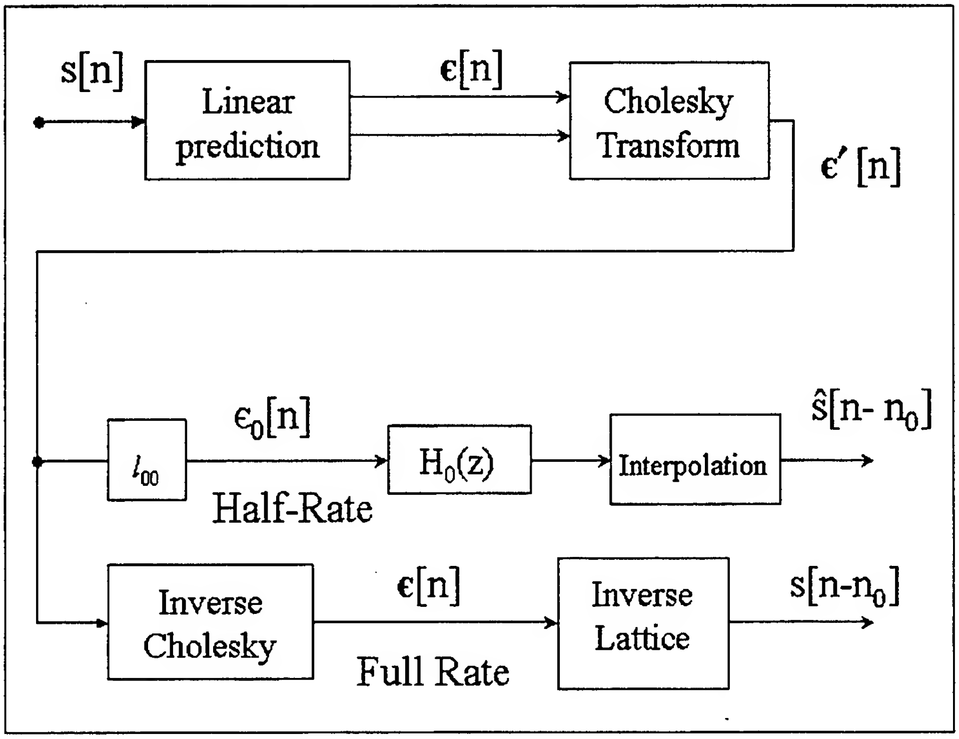

Figure 24. Possible Application of Multirate Linear Prediction

The main focus of this research has been on the development of the theoretical framework. Subsequent research should focus on the applicability of both the multirate Wiener filter as well as the multiresolution model to a diverse number of problems such as target tracking and speech coding. In addition this class of techniques is also attractive in image processing, as an extension to the multidimensional case.

APPENDIX. MATLAB CODES

```
function [h,g,MSE]= MultiWienFIR(Rs,N,M,the,gam,varU,varV,i)
%   Function to find the time-variant FIR Wiener filter for a set of
% orders of N & M, given the ACF Rs[l] of a signal s[n]. The vectors
% 'the' and 'gam' denote FIR filters to which the signal is fed
% before noise destortion.
%
% Syntax: [h,g,MSE]= MultiWienFIR(Rs,N,M,the,gam,varU,varV,i);  where
%         h : The part of the filter for x[n] (N x 1)
%         g : The part of the filter for y[m] (M x 1)
%       MSE : The minimum mean-square error
%        Rs : The ACF from Rs[-N] to Rs[N] as a column vector.
%       N,M : The orders for h,g respectively (N at least 1)
%   the,gam : FIR filters impulse responses (P x 1) & (Q x 1)
% varU,varV : The additive noise variances of u[n] and v[m]
%         i : Choise for even {0} or odd {1} time
%
% Note: If M= 0, we compute the classic Wiener filter

%   An auxiliary function for Thesis.
%                     Dimitrios Koupatsiaris
%                     Last Revision: October 12, 2000

k0= find(Rs == max(Rs) ) ;

rsx= filter( the(end:-1:1),1,Rs ) ;   ksx= find(rsx == max(rsx) ) ;

Rxx= filtfilt(the,1,Rs) ;
tempx= toeplitz( Rxx(k0:-1:k0-N+1),Rxx(k0:k0+N-1).' ) ;
RRxx=  tempx + varU*eye(N) ;

% Reduce to classic FIR Wiener filter
if M == 0,
   rr= rsx(ksx:ksx+N-1) ;
   h= inv(RRxx)*rr ;
   g= [] ;
   MSE= Rs(k0) - h'*rr ;
   return
end
```

```
d= eye(M);   D0= [] ;        % Decimation Matrix
for k= 1:M,
   D0= [ D0 , zeros(M,1) , d(:,k) ] ;
end
D1= rev(D0) ;

switch i
case 0,
   D= D0 ;
case 1,
   D= rev(D0) ;
otherwise,
   error('Wrong choice for time parameter') ;
end

rsy= filter( gam(end:-1:1),1,Rs ) ;  ksy= find(rsy == max(rsy) ) ;

Ryy= filtfilt(gam,1,Rs) ;
tempy= toeplitz( Ryy(k0:-1:k0-2*M+1),Ryy(k0:k0+2*M-1).' ) ;
RRyy = D1*tempy*D1' + varV*eye(M) ;

rxy= filter(the,1,Rs) ;     Rxy= filter( gam,1,rxy(end:-1:1) ) ;
Rxy= Rxy(end:-1:1) ;
kxy= find( abs(Rxy) == max(abs(Rxy)) ) ;

tempxy= toeplitz( Rxy(kxy:-1:kxy-N+1),Rxy(kxy:kxy+2*M-1).' ) ;
RRxy= tempxy*rev(D') ;

RR= [ [RRxx;RRxy'],[RRxy;RRyy] ] ;

r1= rsx(ksx:ksx+N-1) ;
r2= rev(D)*rsy(ksy:ksy+2*M-1) ;

rr= [ r1 ; r2 ] ;

H= inv(RR)*rr ;
h= H(1:N) ;   g= H(N+1:end) ;

MSE= Rs(k0) - r1'*h - r2'*g ;
```

```
function [A,B,Sa,Sb,gam,gama]= ChanLevin(M,R)
% ChanLevin Computes the Levinson Recursion for an M-Channel signal
%              given a ( M*P x M*P ) matrix R. The last matrix is a
%              block Toeplitz and represents the Autocorrelation Matrix
%              for the M-channel signal.
%  Syntax: [A,B,Sa,Sb,gam,gama]= ChanLevin(M,R)             where
%          A : The set of forward PEF coefficients (M*P x M)
%          B : The set of backward PEF coefficients (M*P x M)
%         Sa : The forward covariance of prediction error (M x M)
%         Sb : The backward covariance of prediction error (M x M)
%        gam : The reflection coefficients formed as a cell
% array of order P x 2 { GAMa,GAMb}
%       gama : The reflection coefficients formed as matrix [GAMa,GAMb]

% Script file for THESIS work
%                   Advisors :      C.W. Therrien - R. Cristi
%                   Student  :      Dimitrios Koupatsiaris
%                                   Last Revision: November 03, 2000

P= round(size(R,2)/M)-1 ;
A= eye(M); B= eye(M);    % Initialize A & B
Sa= R(1:M,1:M);   Sb= Sa;            % Initialize covariance of error
gam= cell(P,2) ;   gama= [] ;

for  i= 1:P,
   Amat= [];   Bmat= [];   temp= [];
   for j= 0:round(size(A,1)/M)-1,
      Amat= [Amat ; A(end-M*(j+1)+1:end-M*j,:)] ;
      Bmat= [Bmat ; B(end-M*(j+1)+1:end-M*j,:)] ;
      temp= [ temp, R(1:M,M*(j+1)+1:M*(j+2))' ] ;
   end
   Dp= temp*Amat ;
   GAMa= inv(Sb)*Dp ;
   GAMb= inv(Sa)*Dp' ;
   gam{i,1}= GAMa ;    gam{i,2}= GAMb ;
   gama= [gama ; [GAMa,GAMb] ] ;
   A= [A;zeros(M)] - [zeros(M);Bmat]*GAMa ;
   B= [B;zeros(M)] - [zeros(M);Amat]*GAMb ;
   Sa= Sa*( eye(M) - GAMb*GAMa ) ;
   Sb= Sb*( eye(M) - GAMa*GAMb ) ;
end
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Burrus, C.S., Gopinah, R.A., and Guo, H. *Wavelets and Wavelet Transforms*, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1998.

[2] Therrien, C.W. *Discrete Random Signals and Statistical Signal Processing*, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1992.

[3] Proakis, J.G. and Manolakis, D.G. *Digital Signal Processing*, Prentice Hall, Inc., Upper Saddle River, New Jersey, 1996.

[4] Chou, K.C., Willsky, A.S., and Benveniste, A. "Multiscale Recursive Estimation, Data Fusion and Regularization," *IEEE Transactions on Automatic Control*, 39(3):464–478, March 1994.

[5] Chou, K.C., Willsky, A.S., and Nikoukhah, R. "Multiscale Systems, Kalman Filters and Riccati Equations," *IEEE Transactions on Automatic Control*, 39(3):479–492, March 1994.

[6] Vaidyanathan, P.P. *Multirate Systems and Filter Banks*, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1993.

[7] Basseville, M., Benveniste, A., Chou, K.C., Golden, S.A., Nikoukhah, R., and Willsky, A.S. "Modeling and Estimation of Multiresolution Stochastic Processes," *IEEE Transactions on Information Theory*, 38(2):766–784, March 1992.

[8] Cristi, R. and Tummala, M. "Multirate, Multiresolution Recursive Kalman Filter," *Signal Processing*, November 2000.

[9] Ohno, S. and Sakai, H. "Optimization of Filter Banks Using Cyclostationary Spectral Analysis," *IEEE Transactions on Signal Processing*, 44(11):2718–2725, November 1996.

[10] Strang, G. *Linear Algebra and its Applications*, Harcourt Brace Jovanovich, Orlando, Florida, 1988.

[11] Nuttall, A. H. "Multivariate Linear Predictive Spectral Analysis Employing Weighted Forward and Backward Averaging: A Generalization of Burg's Algorithm," Technical Report 5501, Naval Underwater Systems Center, New London, Connecticut, October 1976. (In volume entitled "NUSC Scientific and Engineering Studies, *Spectral Estimation*").

[12] Strand, O. N. "Multichannel Complex Maximum Entropy (Autoregressive) Spectral Analysis," *IEEE Transactions on Automatic Control*, AC-22(4):634–640, August 1977.

[13] Therrien, C. W. "The Analysis of Multichannel Two-Dimensional Random Signals," Technical Report NPS62-87-002, Naval Postgraduate School, Monterey, California, October 1986.

[14] Wiggins, R. A. and Robinson, E. A. "Recursive Solution to the Multichannel Filtering Problem," *Journal of Geophysical Resources*, 70:1885–1891, April 1965.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Road., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Chairman, Code EC 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

4. Prof. Charles W. Therrien, Code EC/Ti 2
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

5. Prof. Roberto Cristi, Code EC/Cx 2
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

6. Prof. Xiaoping Yun, Code EC/Yx 2
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

7. Embassy of Greece 2
   Office of Naval Attaché
   2228 Massachusetts Avenue, N.W.
   Washington, DC 20008

8. Dimitrios A. Koupatsiaris, Lt. Hellenic Navy 2
   Pelloponisou 27
   Agia Paraskevi, ATTIKI 153-41
   GREECE (HELLAS)